Technologien für die intelligente Automation
Technologies for Intelligent Automation



Volker Lohweg *Hrsg.*

Bildverarbeitung in der Automation

Ausgewählte Beiträge des Jahreskolloquiums BVAu 2022

OPEN ACCESS

Springer Vieweg

Technologien für die intelligente Automation

Technologies for Intelligent Automation

Band 17

Reihe herausgegeben von

inIT – Institut für industrielle Informationstechnik, Technische Hochschule Ostwestfalen-Lippe, Lemgo, Deutschland

Ziel der Buchreihe ist die Publikation neuer Ansätze in der Automation auf wissenschaftlichem Niveau, Themen, die heute und in Zukunft entscheidend sind, für die deutsche und internationale Industrie und Forschung. Initiativen wie Industrie 4.0, Industrial Internet oder Cyber-physical Systems machen dies deutlich. Die Anwendbarkeit und der industrielle Nutzen als durchgehendes Leitmotiv der Veröffentlichungen stehen dabei im Vordergrund. Durch diese Verankerung in der Praxis wird sowohl die Verständlichkeit als auch die Relevanz der Beiträge für die Industrie und für die angewandte Forschung gesichert. Diese Buchreihe möchte Lesern eine Orientierung für die neuen Technologien und deren Anwendungen geben und so zur erfolgreichen Umsetzung der Initiativen beitragen.

Volker Lohweg
(Hrsg.)

Bildverarbeitung in der Automation

Ausgewählte Beiträge des Jahreskolloquiums BVAu 2022

Springer Vieweg

Hrsg.
Volker Lohweg
inIT – Institut für industrielle
Informationstechnik, Technische Hochschule
Ostwestfalen-Lippe
Lemgo, Deutschland

ISSN 2522-8579 ISSN 2522-8587 (electronic)
Technologien für die intelligente Automation
ISBN 978-3-662-66768-2 ISBN 978-3-662-66769-9 (eBook)
https://doi.org/10.1007/978-3-662-66769-9

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Springer Vieweg

Planung/Lektorat: Alexander Grün
Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer-Verlag GmbH, DE und ist ein Teil von Springer Nature.
Die Anschrift der Gesellschaft ist: Heidelberger Platz 3, 14197 Berlin, Germany

Preface

The present conference proceedings "Bildverarbeitung in der Automation" (BVAu, Image Processing in Automation) of the inIT – Institute Industrial IT are based on the contributions of the scientific annual colloquium BVAu 2022.

Industrial image processing continues to establish itself as a key technology in manufacturing companies. Image processing is indispensable with regard to quality assurance through optical measurement strategies, machine conditioning and product analysis as well as human-computer interaction. In this regard, both classical and machine-learning image processing and pattern recognition methods are capable of amazing performance today. In industrial automation, they are subject to more demanding requirements, such as real-time capability or robustness, and have to be operational under demanding conditions, e.g. resource-limited hardware or scarce data.

The main topics of this year's BVAu 2022 are quality inspection, intelligent image processing despite scarce data and the application of AI methods to various industrial application examples. The contributions to this conference proceedings deal in depth with the methods of data augmentation, training on synthetic data, deep learning and anomaly detection. Applications range from industrial production to banknote printing and laboratory experiments.

The authors show the thematic diversity of current challenges in state-of-the-art image processing algorithms. With their contributions, they advance the applicability and robustness of intelligent image processing for industrial automation. We hope that you enjoy reading this publication.

Lemgo — Volker Lohweg
November 2022 — Christoph-Alexander Holst

Organisation

Image Processing in Automation – BVAu 2022

The biennal colloquium Image Processing in Automation is a panel for science and industry covering both technical and scientific issues regarding industrial image processing and pattern recognition. The colloquium is organised by the inIT – Institute Industrial IT of the Technische Hochschule Ostwestfalen-Lippe in Lemgo, Germany.

Conference Chair

Prof. Dr. Volker Lohweg	inIT – Institute Industrial IT, Technische Hochschule Ostwestfalen-Lippe

Program Committee

Prof. Dr. Ulrich Büker	Technische Hochschule Ostwestfalen-Lippe
Prof. Dr. Helene Dörksen	inIT – Institute Industrial IT, Technische Hochschule Ostwestfalen-Lippe
Dr.-Ing. Olaf Enge-Rosenblatt	Fraunhofer IIS, Division Engineering of Adaptive Systems EAS
Prof. Dr.-Ing. Diana Göhringer	Technische Universität Dresden
Christoph-Alexander Holst	inIT – Institute Industrial IT, Technische Hochschule Ostwestfalen-Lippe
Prof. Dr.-Ing. Michael Hübner	Brandenburg University of Technology Cottbus-Senftenberg
Eugen Gillich	Koenig & Bauer Banknote Solutions

Prof. Dr. Oliver Niggemann	Helmut Schmidt Universität Hamburg
Prof. Dr.-Ing. Ralf Salomon	Universität Rostock
Prof. Dr. Karl Schaschek	Hochschule der Medien Stuttgart

Organising Comittee

Roland Hildebrand	inIT – Institute Industrial IT, Technische Hochschule Ostwestfalen-Lippe
Benedikt Lücke	
Stephanie Wisser	
Jasmin Zilz	

Contents

Contributors

Shobhit Agarwal Fraunhofer Institute for Production Technology IPT, Aachen, Deutschland

Markus Brüning inIT – Institute Industrial IT, Technische Hochschule Ostwestfalen-Lippe, Lemgo, Deutschland

Helene Dörksen inIT – Institute Industrial IT, Technische Hochschule Ostwestfalen-Lippe, Lemgo, Deutschland

Alexander Fay Helmut-Schmidt-Universität, Hamburg, Deutschland

Holger Flatt Fraunhofer IOSB, Institutsteil für industrielle Automation (IOSB-INA), Lemgo, Deutschland

Theo Gabloffsky University of Rostock, Rostock, Deutschland

Florian Gellert Fraunhofer IOSB, Institutsteil für industrielle Automation (IOSB-INA), Lemgo, Deutschland

Milapji Singh Gill Helmut-Schmidt-Universität, Hamburg, Deutschland

Iris Gräßler Heinz Nixdorf Institute, Paderborn, Deutschland

Alexander Hawlitschka University of Rostock, Rostock, Deutschland

Michael Hieb Heinz Nixdorf Institute, Paderborn, Deutschland

Marvin Höhner Helmut-Schmidt-Universität, Hamburg, Deutschland

Florian Hufen Fraunhofer IOSB, Institutsteil für industrielle Automation (IOSB-INA), Lemgo, Deutschland

Lars Leyendecker Fraunhofer Institute for Production Technology IPT, Aachen, Deutschland

Dorit Merhof Institute of Image Analysis and Computer Vision, University of Regensburg, Regensburg, Deutschland

Maximilian Motz Fraunhofer Institute for Production Technology IPT, Aachen, Deutschland

Oliver Rippel Institute of Imaging & Computer Vision, RWTH Aachen University, Aachen, Deutschland

Daniel Roesmann Heinz Nixdorf Institute, Paderborn, Deutschland

Ralf Salomon University of Rostock, Rostock, Deutschland

Robert H. Schmitt Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen, Aachen, Deutschland

Dennis Sprute Fraunhofer IOSB, Institutsteil für industrielle Automation (IOSB-INA), Lemgo, Deutschland

Fabian Stoller Helmut-Schmidt-Universität, Hamburg, Deutschland

Philip Topalis Helmut-Schmidt-Universität, Hamburg, Deutschland

Marc Unverzagt Heinz Nixdorf Institute, Paderborn, Deutschland

Thorben Werner Information Systems and Machine Learning Lab (ISMLL), Hildesheim, Deutschland

Tim Westerhold Fraunhofer IOSB, Institutsteil für industrielle Automation (IOSB-INA), Lemgo, Deutschland

Paul Wunderlich inIT – Institute Industrial IT, Technische Hochschule Ostwestfalen-Lippe, Lemgo, Deutschland

Anomaly Detection for Automated Visual Inspection: A Review

Oliver Rippel and Dorit Merhof

Abstract

Anomaly detection (AD) methods that are based on deep learning (DL) have considerably improved the state of the art in AD performance on natural images recently. Combined with the public release of large-scale datasets that target AD for automated visual inspection (AVI), this has triggered the development of numerous, novel AD methods specific to AVI. However, with the rapid emergence of novel methods, the need to systematically categorize them arises. In this review, we perform such a categorization, and identify the underlying assumptions as well as working principles of DL-based AD methods that are geared towards AVI. We perform this for 2D AVI setups, and find that the majority of successful AD methods currently combines features generated by pre-training DL models on large-scale, natural image datasets with classical AD methods in hybrid AD schemes. Moreover, we give the main advantages and drawbacks of the two identified model categories in the context of AVI's inherent requirements. Last, we outline open research questions, such as the need for an improved detection performance of semantic anomalies, and propose potential ways to address them.

Keywords

Anomaly detection · Automated visual inspection · Deep learning · Quality control

O. Rippel (✉) · D. Merhof
Institute of Imaging & Computer Vision, RWTH Aachen University, Aachen, Deutschland
e-mail: oliver.rippel@lfb.rwth-aachen.de

D. Merhof
e-mail: dorit.merhof@lfb.rwth-aachen.de

V. Lohweg (Hrsg.), *Bildverarbeitung in der Automation*, Technologien für die intelligente Automation 17, https://doi.org/10.1007/978-3-662-66769-9_1

1 Introduction

Anomaly detection (AD) tries to identify instances in data that deviate from a previously defined (or learned) concept of “normality” [1, 2]. In this context, identified deviations are referred to as “anomalies”, and labeled as “anomalous”, whereas data points that conform to the concept of normality are considered “normal”. In the field of computer vision, AD tries to identify anomalous images, and one of its most promising application domains is the automated visual inspection (AVI) of manufactured goods [3–6]. The reason for this is the close match between the properties inherent to the AD problem and the constraints imposed by the manufacturing industry on any AVI system (AVIS):

1. Anomalies are rare events [1, 2]. As a consequence, AD algorithms generally focus on finding a description of the normal state, and require few to no anomalies during training. Viewing defective goods as anomalies and the expected product as the normal state, this matches the limited availability of defective goods when setting up AVISs. Manually collecting and labeling defective goods for training supervised deep learning (DL) methods has furthermore been identified as one of the main cost factors for DL-based AVISs [7], and has to be minimized to achieve economic feasibility.
2. The anomaly distribution is ill-defined [1, 2]. This matches the constraint that all possible defect types an AVIS may encounter during deployment are often unknown during training [4]. Still, AVISs are expected to detect also such unknown defect types reliably.

These two constraints imposed by AD problems in general, and the manufacturing industry in particular, already severely limit the feasibility of supervised, DL-based AVISs. Additionally, two further requirements are imposed by the manufacturing industry on AVISs:

1. AVI methods should not be compute-intensive during training to minimize lead times of product changes. Said product changes are furthermore expected to become more frequent due to a general decrease in lot sizes inherent to industry 4.0 [8].
2. AVI methods need to run in real-time on limited hardware [9].

While the recent success of DL-based AD algorithms has renewed the general interest in AD [2], these two additional constraints have, combined with the release of public datasets [4, 5, 10, 11], led to the development of AD algorithms that are geared specifically towards AVI [5, 12–22]. The emergence of such algorithms, however, calls for their systematic review in order to consolidate findings and to thereby facilitate additional research. To the best of our knowledge, none of the recent reviews focus on AD for AVI, and they often discuss the broader AD field with a focus on natural images instead [2, 23, 24]. In our work, we fill this gap, and review recent advances in AD for AVI. To this end, we first provide brief formal definitions of both AD and anomaly segmentation (AS), and afterwards summarize public datasets. Next, we systematically categorize algorithms

developed for 2D AVI setups, and give their main advantages as well as disadvantages. Last, we outline open research questions, and propose potential ways of addressing them.

2 A Brief Overview of AD/AS

2.1 Formal Definitions

As outlined above, AD is tasked with deciding whether a given test datum is normal or anomalous. More formally speaking, AD is tasked with finding a function $\Phi\colon \mathcal{X} \to y$ that maps from the input space $\mathcal{X}$ to the anomaly score y. It follows that y should be much lower for a normal test datum than for an anomalous test datum. In context of AVI, $\mathcal{X}$ typically consists of 2D images $\vec{x} \in \mathbb{R}^{C \times H \times W}$. Here, H and W specify the height and width of the image $\vec{x}$, and C corresponds to the number of color channels present in the image. For RGB images, $C = 3$, whereas $C = 1$ for grayscale images and $C > 3$ for multi/hyperspectral images. For a more comprehensive definition of AD, see [2].

In addition to AD, AVI is also concerned with AS, i.e. with localizing the anomaly inside $\vec{x}$. AS is thus tasked with finding a function $\Phi\colon \mathcal{X} \to \vec{y}$ that produces an anomaly map $\vec{y} \in \mathbb{R}^{H \times W}$ instead of a scalar anomaly score y. By aggregating $\vec{y}$ appropriately, an image-level anomaly score y can be subsequently derived for AS algorithms.

2.2 Types of Algorithms

There exist 3 types of AD/AS algorithms, which differ in their requirements w.r.t. the training data (see Fig. 1, [1]):

1. **Supervised algorithms**. Supervised algorithms treat the AD/AS problems as imbalanced, binary classification/segmentation problems. As such, they require a fully labeled dataset that contains both normal and anomalous images for training. Sampling the anomaly distribution furthermore induces a significant bias [25], also for AVI [12].
2. **Semi-supervised algorithms**. As opposed to supervised algorithms, semi-supervised algorithms require only a dataset of labeled normal images for training. Semi-supervised approaches commonly make use of the *concentration assumption* [2], i.e. the assumption that the normal data distribution can be bounded inside a given feature space. Examples here would be neighborhood/prototype [15, 26] or density-based approaches [12, 14, 16]. Other approaches such as autoencoders (AEs) [27] use the *concentration assumption* in a more indirect manner: They try to train models that are well-behaved only on the manifold of the normal data distribution constructed in the input domain, i.e. the raw images. Thereby, they exploit the observation that DL models fail at generalizing to samples that are different from the training dataset [28]. The majority of proposed AD/AS approaches are semi-supervised.

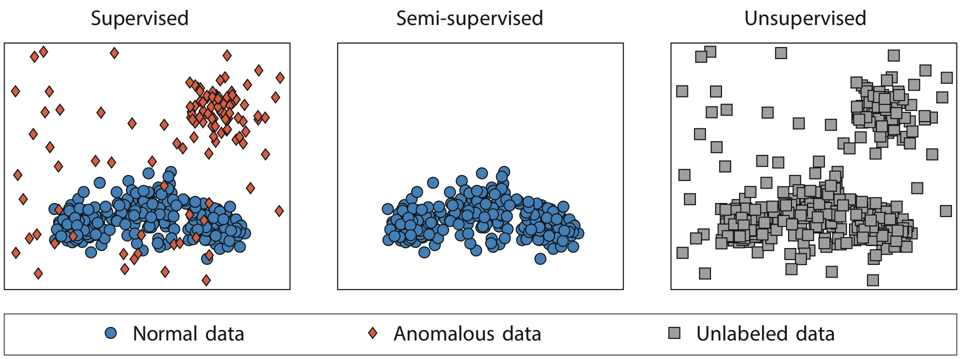


Fig. 1 The three types of AD/AS algorithms. While supervised approaches require both labeled anomalies and normal data for training, semi-supervised approaches use normal data only. Unsupervised approaches work on unlabeled data, and make assumptions about the normal and anomaly distribution.

3. **Unsupervised algorithms**. As opposed to supervised and semi-supervised approaches, unsupervised approaches can work with unlabeled data [29, 30]. To do so, they combine the *concentration assumption* with the two core assumptions made for anomalies: (I) that anomalies are rare events, and (II) that their distribution is ill-defined (still, a uniform distribution is often assumed).

We note that the terms *semi-supervised* and *unsupervised*, as defined in this review based on [1], are not used consistently throughout literature. For example, *unsupervised* is often misused to refer to the *semi-supervised* setting in AVI [4–6]. Moreover, the term *semi-supervised* has also been used to refer to a partially labeled dataset, where labeled anomalies may also be present and used for training [31]. We believe that a consistent use of terminology which conforms with its historical definition [1] would facilitate a more intuitive understanding of research in AD for AVI.

2.3 Anomaly Types

In previous literature, three anomaly types are distinguished [2]:

1. **Point anomalies**, which are instances that are anomalous on their own, e.g. a scratch.
2. **Contextual anomalies**, which are instances that are anomalous only in a specific context. A scratch, for example, might only be considered an anomaly if it lies on a cosmetic surface or otherwise impairs the product's function.
3. **Group anomalies**, which are several data points that, as a whole, form an anomaly. Group anomalies may also be contextual, and are rare in AVI.

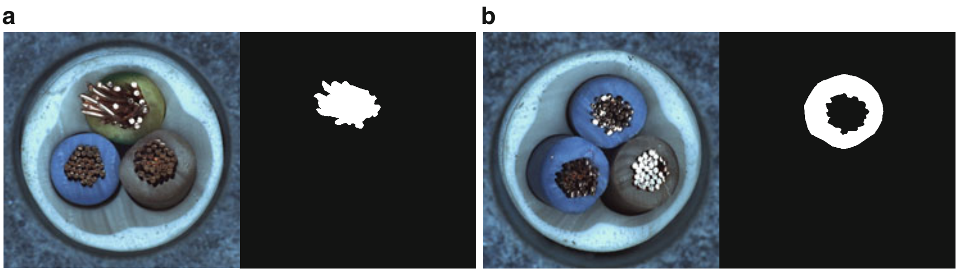


Fig. 2 Textural vs. semantic anomalies. (**a**) shows a textural anomaly, whereas (**b**) shows a semantic anomaly. Images are taken from MVTec AD [4].

Complementary to these categories, anomalies have recently been partitioned based on the degree of semantic understanding required to detect them [2]. In particular, anomalies are divided into low-level, textural anomalies, and high-level, semantic anomalies (see Fig. 2 for an example). Detecting semantic anomalies is generally more difficult than detecting textural anomalies [2, 5], as learning semantically meaningful feature representation is inherently more difficult. Convolutional neural networks (CNNs), for example, exhibit a significant texture bias [32, 33]. Furthermore, we note that synonymous nomenclature was introduced in AD for AVI recently [5], where textural anomalies correspond to structural anomalies, and semantic anomalies correspond to logical anomalies. We again stress the importance of using terminology consistently, and stick to the terms textural/semantic anomalies due to their more intuitive understanding.

2.4 Evaluating AD/AS Performance

Since AD and AS can be viewed as binary problems, corresponding evaluation measures are commonly used to evaluate their performance. Specifically, the area under the receiver operating characteristic (ROC) curve (AUROC) and the area under the precision-recall (PR) curve (AUPR) are employed. It should be noted that AUPR is better-suited for evaluating imbalanced problems such as AS [34]. Recently, the per-region-overlap (PRO) curve was proposed specifically for AS in AVI [13]. As opposed to the pixel-wise AUROC/AUPR, the area under the PRO curve is used for measuring an algorithm's ability to detect all individual anomalies present in an image. Since the PRO curve does not take false positive (FP) predictions into account, it is constructed up to a preset false positive rate (FPR), and the most commonly used cut-off value is 30% FPR.

We note that all the above evaluation measures focus on an algorithm's general capability of solving the AD/AS problem. Thereby, the difficulty of selecting the optimal threshold t for $y/\vec{y}$ is sidestepped by iterating over all possible thresholds t. Finding the optimal value for t, and finding it ideally with normal images only, is a promising avenue for future research [3].

3 AD/AS for 2D AVI

3.1 Datasets

Recent advances in AD/AS for AVI were spurred by the public release of suitable datasets that depict manufactured goods such as screws (objects) or fabrics (textures). We give an overview of them in Table 1, and observe the following:

1. In none of the datasets, anomalies are rare events. In fact, global prevalence ranges from 10%–30%, and prevalences commonly reach >50% in the pre-defined test sets. Furthermore, total dataset sizes are relatively small compared to the throughput of a deployed AVIS, and datasets are thus at risk of not fully capturing the normal data distribution. Fully recapitulating the normal data distribution, however, is crucial to achieve high true positive rates (TPRs) at low FPRs, a requirement imposed by the rarity of anomalies/defects. Thus, developed AD/AS methods might not transfer as well to industry. To mitigate this, dataset sizes should be further increased, and an effort should be made to sample the normal data distribution as representatively as possible.
2. Almost no dataset specifies the anomaly type. In fact, only MVTec LOCO AD distinguishes between textural and semantic anomalies, and even MVTec LOCO AD does not differentiate between point, contextual, and group anomalies. Furthermore, MVTec AD, BTAD and MTD contain mostly textural anomalies. Together, this limits research aimed at detecting semantic anomalies in AVI, and additional datasets are required.
3. All datasets contain images that can be cast to the RGB format. Moreover, most of the goods contained in MVTec AD and MVTec LOCO AD are relatively similar in appearance to natural images.

Table 1 Public AD datasets for 2D AVI. In addition to the defect labels (e.g. scratch), we also denote whether the anomaly labels are given (i.e. textural vs. semantic). Abbreviations: Obj. = Objects, Text. = Textures, Norm. = Normal, Anom. = Anomalous, C = # Channels, M.c.s.l. = Multi-class single label.

Dataset	Goods			Images				Label		Mask
	Obj.	Text.	Total	Norm.	Anom.	Total	C	Defect	Anomaly	
MVTec AD [4]	10	5	15	4096	1258	5354	3	M.c.s.l	✗	✓
BTAD [10]	2	1	3	2250	290	2540	3	Binary	✗	✓
MTD [11]	0	1	1	952	392	1344	1	M.c.s.l	✗	✓
MVTec LOCO AD [5]	5	0	5	2651	993	3644	3	M.c.s.l	✓	✓

3.2 Methods

In general, developed AD/AS methods can be categorized into those that train complex models and their feature representations in an end-to-end manner, and hybrid approaches that leverage feature representations of DL models pre-trained on large-scale, natural image datasets, but leave them unchanged.

3.2.1 Training Complex Models in an End-to-End Manner.

Methods that train complex models in an end-to-end manner tend to pursue either AEs [27] or knowledge distillation (KD) [13]. Both approaches are based on the assumption that the trained DL model is well-behaved only on images that originate from the normal data distribution. For the AE, this means that the image reconstruction fails for anomalies, whereas for KD, this means that the regression of the teacher's features by the student network fails. While the AE can be easily applied to multi/hyperspectral images [35], KD-based approaches are limited by their need for a suitable teacher model. Since CNNs pre-trained on ILSVRC2012 (a subset of ImageNet [36]) are commonly used as teacher models, this limits KD approaches to images that are castable to the RGB image format used in ImageNet, i.e. RGB or grayscale images. While a randomly initialized CNN might potentially be used as the teacher to circumvent this problem (similar approaches have been pursued successfully in reinforcement learning [37]), its efficacy has not yet been demonstrated for AVI.

As an alternative to AE and KD, the *concentration assumption* can be used to formulate learning objectives such as the patch support vector data description [38], which directly learn feature representations where the normal data is concentrated/clustered around a fixed point. Anomalies are then expected to have a larger distance to the cluster center than normal data.

The main advantage of methods that train complex models in an end-to-end manner is their applicability to any data type, including multi/hyperspectral images. For their main drawbacks, it needs to be stated that training these methods is compute-intensive, and they thus do not conform with the requirement of low/limited training effort imposed by the manufacturing industry. Furthermore, these methods tend to produce worse results than hybrid approaches on RGB-castable images. As a potential explanation, it has been hypothesized that discriminative features are inherently difficult to learn from scratch using normal data only [21]. Moreover, it was shown that AEs tend to generalize to anomalies in AVI [27]. To improve results, two approaches are currently pursued in literature: (I) Initializing the method with a model that was pre-trained on a large-scale natural image dataset [39]. However, this restricts approaches to grayscale/RGB images due to a lack of large-scale multi/hyperspectral image datasets. Furthermore, its effectiveness is limited by catastrophic forgetting [22, 40], which, in AD, refers to a loss of initially present, discriminative features. Therefore, this technique is often combined with the second approach, where (II) surrogates for the anomaly distribution are provided via anomaly synthesis [31, 41]. This requires either access to representative anomalies as a basis for synthesis,

or an exhaustive understanding of the underlying manufacturing process and the visual appearance of occurring defects. Thus, anomaly synthesis violates the assumption of an ill-defined anomaly distribution in the same manner as supervised approaches, and is expected to incur a similar bias.

3.2.2 Hybrid Approaches.

At their core, hybrid approaches assume that feature representations generated by training DL models on large-scale, natural image datasets can be transferred to achieve AD in AVI. Specifically, they assume that discriminative features have already been generated, and restrict themselves to finding a description of normality in said features. To this end, they employ three different techniques, all of which are classical AD approaches that are based on the *concentration assumption*: (I) Generative approaches explicitly model the probability density function (PDF) of the normal data distribution inside the pre-trained feature representations. Both unconstrained PDFs (i.e. via normalizing flows) [16, 42] and constrained PDFs (e.g. by assuming a Gaussian distribution) [12, 14] have been used. (II) Classification-based approaches fit binary classification models such as the one-class support vector machine (SVM) to the pre-trained feature representations [43]. (III) Neighborhood/Prototype-based approaches employ k-NN algorithms or variations thereof to implicitly approximate the PDF of the normal data distribution [15, 26].

A main advantage of hybrid approaches lies in their outstanding performance: All approaches that achieved state-of-the-art AD performance on MVTec AD so far are hybrid approaches (see Fig. 3). Furthermore, hybrid approaches are in general not compute-intensive during training, as they do not train complex DL models. For example, "training" a Gaussian AD model consists of extracting the pre-trained features for the training dataset, followed by the numeric computation of $\vec{\mu}$ and $\vec{\Sigma}$ [12]. Moreover, hybrid approaches are fast, as state-of-the-art, lightweight classification CNNs can be used

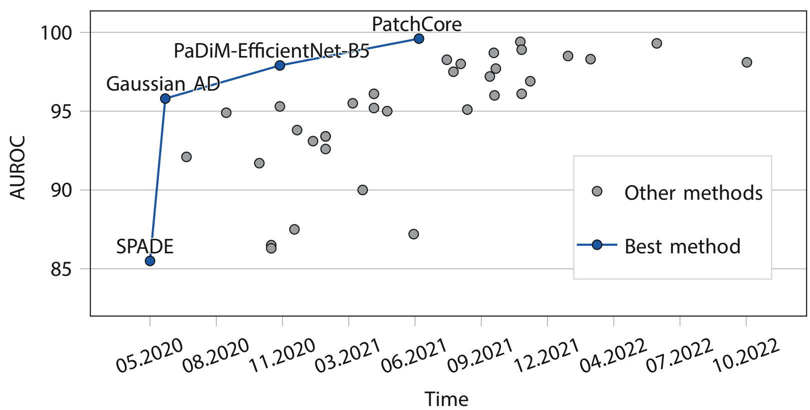


Fig. 3 Temporal progression of the state of the art in AD performance on MVTec AD. Data was sourced from https://paperswithcode.com/about on 26.08.2022.

as feature extractors. Thereby, they fit the requirements of the manufacturing industry extremely well.

The main disadvantage of hybrid approaches lies in their core assumption: If the feature representations of the underlying, pre-trained feature extractor/model are simply not discriminative to the specific AVI problem at hand, hybrid approaches will automatically fail. However, hybrid approaches have been successfully applied even to AVI setups which produce images that differ from the natural image distribution [15, 18, 42]. To nonetheless mitigate this disadvantage, the diversity of available, pre-trained feature extractors should be increased. There are two straightforward ways to do this: (I) Using datasets other than ILSVRC2012 for pre-training, and (II) Using different model architectures, and even computer vision tasks, for pre-training. Both influence general transfer learning performance [44], and initial work indicates they might be beneficial also for AD/AS in AVI [20]. The second disadvantage of hybrid approaches is their limitation to images that can be cast to the RGB format, which is directly due to the underlying feature extractors that are trained on natural image datasets.

4 Open Research Questions

First, the detection performance of semantic anomalies needs to be improved further. This would facilitate the application of developed algorithms to even more sophisticated AVI tasks. Here, hybrid approaches can directly benefit from advances in DL which yield more semantically meaningful feature representations. For example, vision transformers were recently shown to possess as smaller texture bias than CNNs [45], and could thus potentially be used as feature extractors. Second, the bias incurred by sampling the anomaly distribution needs to be decreased. A potential way of achieving this would be to explicitly incorporate the assumptions made for the anomaly distribution into the corresponding learning objectives. Third, methods that facilitate setting the working point of AD/AS methods in an automated manner are needed. As these would ideally rely on normal data only, they could aim at achieving specific target FPRs, e.g. via bootstrapping and model ensembling. Fourth, AD/AS methods that are less compute-intensive during training are required for multi/hyperspectral images to meet the requirements imposed by the manufacturing industry. Here, public datasets are expected to facilitate progress, similar as was observed for RGB images. Fifth, AD/AS methods are required for 3D AVI tasks. A first dataset was published recently [6], and we expect for hybrid approaches that rely on models pre-trained on large-scale datasets to also achieve strong performance here [17].

While not an open research questions specific to AVI, anomalies have recently been clustered successfully based on their visual appearance [46]. Together with the empirical success of anomaly synthesis/supervised AD [31, 41, 47], this indicates that the commonly made assumption that anomalies follow a uniform distribution might not be true. This aspect thus warrants additional research.

5 Conclusion

In our work, we have reviewed recent advances in AD/AS for AVI. We have provided a brief definition of AD/AS, and gave an overview of public datasets and their limitations. Moreover, we identified two general categories of AD/AS approaches for AVI, and gave their main advantages as well as disadvantages when considering the constraints and requirements imposed by the manufacturing industry. Last, we identified open research questions, and outlined potential ways of addressing them. We expect our review to facilitate additional research in AD/AS for AVI.

References

1. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: A survey. Acm Comput Surv (CSUR) 41(3):1–58
2. Ruff L, Kauffmann JR, Vandermeulen RA, Montavon G, Samek W, Kloft M, Dietterich TG, Müller KR (2021) A unifying review of deep and shallow anomaly detection. Proc IEEE 109(5):756–795
3. Bergmann P, Fauser M, Sattlegger D, Steger C (2019) MVTec AD – a comprehensive real-world dataset for unsupervised anomaly detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
4. Bergmann P, Batzner K, Fauser M, Sattlegger D, Steger C (2021) The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection. Int J Comput Vis. https://doi.org/10.1007/s11263-020-01400-4
5. Bergmann P, Batzner K, Fauser M, Sattlegger D, Steger C (2022) Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. Int J Comput Vis. https://doi.org/10.1007/s11263-022-01578-9
6. Bergmann P, Jin X, Sattlegger D, Steger C (2022) The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. In: Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications – Volume 5: VISAPP. SciTePress, Setúbal, S 202–213
7. Dai W, Mujeeb A, Erdt M, Sourin A (2018) Towards automatic optical inspection of soldering defects. In: 2018 International Conference on Cyberworlds (CW), S 375–382
8. Brettel M, Friederichsen N, Keller M, Rosenberg M (2014) How virtualization, decentralization and network building change the manufacturing landscape: An industry 4.0 perspective. Int J Inf Commun Eng 8(1):37–44
9. Li L, Ota K, Dong M (2018) Deep learning for smart industry: Efficient manufacture inspection system with fog computing. IEEE Trans Ind Informatics 14(10):4665–4673
10. Mishra P, Verk R, Fornasier D, Piciarelli C, Foresti GL (2021) VT-ADL: A vision transformer network for image anomaly detection and localization. In: 30th IEEE/IES International Symposium on Industrial Electronics (ISIE). Kyoto, Japan
11. Huang Y, Qiu C, Yuan K (2020) Surface defect saliency of magnetic tile. Vis Comput 36(1):85–96
12. Rippel O, Mertens P, König E, Merhof D (2021) Gaussian anomaly detection by modeling the distribution of normal data in pretrained deep features. IEEE Trans Instrum Meas 70:1–13

13. Bergmann P, Fauser M, Sattlegger D, Steger C (2020) Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, S 4183–4192
14. Defard T, Setkov A, Loesch A, Audigier R (2021) Padim: A patch distribution modeling framework for anomaly detection and localization. In: Del Bimbo A, Cucchiara R, Sclaroff S, Farinella GM, Mei T, Bertini M, Escalante HJ, Vezzani R (Hrsg) Pattern Recognition. ICPR International Workshops and Challenges. Springer, Cham, S 475–489
15. Roth K, Pemula L, Zepeda J, Schölkopf B, Brox T, Gehler P (2022) Towards total recall in industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), S 14318–14328
16. Gudovskiy D, Ishizaka S, Kozuka K (2022) Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), S 98–107
17. Bergmann P, Sattlegger D (2022) Anomaly detection in 3d point clouds using deep geometric descriptors (arXiv preprint arXiv:2202.11660)
18. Rippel O, Haumering P, Brauers J, Merhof D (2021) Anomaly detection for the automated visual inspection of pet preform closures. In: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), S 1–7
19. Rippel O, Müller M, Merhof D (2020) GAN-based defect synthesis for anomaly detection in fabrics. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Bd. 1, S 534–540
20. Rippel O, Merhof D (2021) Leveraging pre-trained segmentation networks for anomaly segmentation. In: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), S 1–4
21. Rippel O, Mertens P, Merhof D (2021) Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In: 2020 25th International Conference on Pattern Recognition (ICPR), S 6726–6733
22. Rippel O, Chavan A, Lei C, Merhof D (2022) Transfer learning gaussian anomaly detection by fine-tuning representations. In: Proceedings of the 2nd International Conference on Image Processing and Vision Engineering – IMPROVE. INSTICC, SciTePress, Setúbal, S 45–56
23. Chalapathy R, Chawla S (2019) Deep learning for anomaly detection: A survey (arXiv preprint arXiv:1901.03407)
24. Pang G, Shen C, Cao L, Hengel AVD (2021) Deep learning for anomaly detection: A review. ACM Comput Surv 54(2):1–38
25. Ye Z, Chen Y, Zheng H (2021) Understanding the effect of bias in deep anomaly detection. In: Zhou ZH (Hrsg) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19–27 August 2021, S 3314–3320
26. Cohen N, Hoshen Y (2020) Sub-image anomaly detection with deep pyramid correspondences (arXiv preprint arXiv:2005.02357)
27. Zavrtanik V, Kristan M, Skocaj D (2021) Draem – a discriminatively trained reconstruction embedding for surface anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), S 8330–8339
28. Hendrycks D, Basart S, Mu N, Kadavath S, Wang F, Dorundo E, Desai R, Zhu T, Parajuli S, Guo M, Song D, Steinhardt J, Gilmer J (2021) The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), S 8340–8349
29. Cordier A, Missaoui B, Gutierrez P (2022) Data refinement for fully unsupervised visual inspection using pre-trained networks (arXiv preprint arXiv:2202.12759)

30. Yoon J, Sohn K, Li CL, Arik SO, Lee CY, Pfister T (2022) Self-supervise, refine, repeat: Improving unsupervised anomaly detection. Transactions on machine learning research
31. Liznerski P, Ruff L, Vandermeulen RA, Franks BJ, Kloft M, Müller KR (2021) Explainable deep one-class classification. In: International Conference on Learning Representations
32. Geirhos R, Rubisch P, Michaelis C, Bethge M, Wichmann FA, Brendel W (2019) Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations
33. Hermann KL, Chen T, Kornblith S (2020) The origins and prevalence of texture bias in convolutional neural networks. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (Hrsg) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual
34. Davis J, Goadrich M (2006) The relationship between precision-recall and roc curves. In: Proceedings of the 23rd International Conference on Machine learning, S 233–240
35. Ma N, Peng Y, Wang S, Liu D (2018) Hyperspectral image anomaly targets detection with online deep learning. In: 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). IEEE, New York, S 1–6
36. Deng J, Dong W, Socher R, Li L, Li K, Fei-Fei L (2009) ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, S 248–255
37. Burda Y, Edwards H, Storkey A, Klimov O (2019) Exploration by random network distillation. In: International Conference on Learning Representations
38. Yi J, Yoon S (2020) Patch svdd: Patch-level svdd for anomaly detection and segmentation. In: Proceedings of the Asian Conference on Computer Vision (ACCV)
39. Venkataramanan S, Peng KC, Singh RV, Mahalanobis A (2020) Attention guided anomaly localization in images. In: Vedaldi A, Bischof H, Brox T, Frahm JM (Hrsg) Computer Vision – ECCV 2020. Springer, Cham, S 485–503
40. Deecke L, Ruff L, Vandermeulen RA, Bilen H (2020) Deep anomaly detection by residual adaptation (arXiv preprint arXiv:2010.02310)
41. Li CL, Sohn K, Yoon J, Pfister T (2021) Cutpaste: Self-supervised learning for anomaly detection and localization. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), S 9659–9669
42. Rudolph M, Wandt B, Rosenhahn B (2021) Same same but differnet: Semi-supervised defect detection with normalizing flows. In: Winter Conference on Applications of Computer Vision (WACV)
43. Andrews J, Tanay T, Morton EJ, Griffin LD (2016) Transfer representation-learning for anomaly detection. JMLR, New York
44. Mensink T, Uijlings J, Kuznetsova A, Gygli M, Ferrari V (2021) Factors of influence for transfer learning across diverse appearance domains and task types. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, S 1–1
45. Naseer MM, Ranasinghe K, Khan SH, Hayat M, Shahbaz Khan F, Yang MH (2021) Intriguing properties of vision transformers. In: Advances in Neural Information Processing Systems 34
46. Sohn K, Yoon J, Li CL, Lee CY, Pfister T (2021) Anomaly clustering: Grouping images into coherent clusters of anomaly types (arXiv preprint arXiv:2112.11573)
47. Ruff L, Vandermeulen RA, Franks BJ, Müller KR, Kloft M (2020) Rethinking assumptions in deep anomaly detection (arXiv preprint arXiv:2006.00339)

Bewertungsmetrik für die Bildqualität bei automatisierten optischen Inspektionsanwendungen

Philip Topalis, Marvin Höhner, Fabian Stoller, Milapji Singh Gill und Alexander Fay

Zusammenfassung

Die Sicherstellung qualitativ hochwertiger Bilder in der automatisierten optischen Inspektion (AOI) auf der Basis von Bildverarbeitungsmethoden ist eine notwendige Voraussetzung, um sinnvolle Entscheidungen zu treffen. Verschiedene Einflussfaktoren können sich jedoch negativ auf diese Aufgabe auswirken und zu ungeeigneten Bildern führen. Der Systemaufbau, die Charakteristika des zu untersuchenden Objekts, wie beispielsweise Unregelmäßigkeiten oder Muster, sowie die Durchführung der Untersuchung sind nur einige Beispiele von vielen. In Anwendungsfällen, in denen eine hohe Bildqualität aufgrund der genannten Einflussfaktoren nicht sichergestellt werden kann, ist es erforderlich, die erfassten Bilddaten automatisch hinsichtlich ihrer Eignung für eine robuste automatisierte optische Inspektion zu bewerten. Zu diesem Zweck müssen geeignete Bewertungsmetriken verglichen und entsprechend den spezifischen Anforderungen des Anwendungsfalls ausgewählt werden. In diesem Beitrag wird eine in den laufenden Betrieb eines AOI-Systems integrierte Bildqualitätsbewertung vorgestellt. Zu diesem Zweck wird im Prozess zwischen der Bilderfassung und der Bildauswer-

P. Topalis (✉) · M. Höhner · F. Stoller · M. S. Gill · A. Fay
Helmut-Schmidt-Universität, Hamburg, Deutschland
E-Mail: philip.topalis@hsu-hh.de

M. Höhner
E-Mail: marvin.hoehner@hsu-hh.de

F. Stoller
E-Mail: fabian.stoller@hsu-hh.de

M. S. Gill
E-Mail: milapji.gill@hsu-hh.de

A. Fay
E-Mail: alexander.fay@hsu-hh.de

V. Lohweg (Hrsg.), *Bildverarbeitung in der Automation*, Technologien für die intelligente Automation 17, https://doi.org/10.1007/978-3-662-66769-9_2

tung ein zusätzliches Modul integriert, das in der Lage ist, nicht verwertbare Bilder zu detektieren. Die Anwendung wird anhand eines Demonstrators gezeigt, der Erkenntnisse für die konkrete Umsetzung von AOI-Aufgaben liefern soll.

Schlüsselwörter

Bildverarbeitung · Qualitätskontrolle · Automatisierte Optische Inspektion · Bewertungsmetrik

1 Einleitung

Ein qualitativ hochwertiges Produkt ist ein wesentliches Unterscheidungsmerkmal auf wettbewerbsorientierten Märkten, insbesondere bei sicherheitsrelevanten Anwendungen [1, 2] . Selbst kleinste Defekte können zu Beanstandungen, Funktionseinbußen oder sogar Funktionsausfällen führen. Werden diese Defekte erst im späteren Verlauf des Fertigungsprozesses oder erst im Betrieb erkannt, können diese hohe Kosten verursachen. Eine sorgfältige und zuverlässige Qualitätskontrolle der Produkte ist somit unerlässlich. Im Bereich der optischen Qualitätskontrolle haben Methoden der Bildverarbeitung und des Maschinellen Lernens ihr Potenzial zur Beschleunigung und Verbesserung der Qualitätskontrolle in verschiedenen Bereichen bewiesen [1]. Dennoch werden gerade in der Automobilproduktion und der Wartung von Flugzeugkomponenten (z. B. Avionik-Komponenten wie Flugsteuerungen) hohe Qualitätsanforderungen gestellt. Schließlich kann eine Fehlentscheidung weitreichende Folgen für die sichere Nutzung eines Autos oder eines Flugzeugs haben.

Bevor ein solches automatisiertes und datenbasiertes System zur automatisierten optischen Inspektion (AOI) implementiert und in den Betrieb integriert werden kann, muss also eine hohe Qualität der erfassten Bilder gewährleistet sein. Dies stellt angesichts der Vielzahl möglicher Einflussfaktoren eine große Herausforderung dar [3]. In der Literatur existiert eine Vielzahl von Metriken, die zur Bewertung von Bildern herangezogen werden können. Damit eine Metrik zur Bewertung der Bildqualität für die automatische Auswertung in einem AOI-System eingesetzt werden kann, muss diese die spezifischen Anforderungen an den Einsatz in der Qualitätskontrolle erfüllen. Insofern müssen die Anforderungen und Einflussfaktoren zunächst ermittelt werden, um anschließend eine Auswahl einer geeigneten Metrik zu treffen. Aus diesem Grund sollen im Rahmen dieser Arbeit die folgenden Forschungsfragen beantwortet werden, um Akteure in vergleichbaren Einsatzgebieten bei ihrer Entscheidung zu unterstützen:

RQ1) Welche besonderen Anforderungen ergeben sich aus der Aufgabe der AOI in der Qualitätskontrolle im Automobilbau bzw. der Automobil-Zulieferindustrie und in der Instandhaltung von Flugzeugkomponenten an die zu berücksichtigenden Metriken?

RQ2) Welche Metriken sind nach diesen Anforderungen für die Aufgabe der Qualitätskontrolle in den betrachteten Anwendungsfällen geeignet?
RQ3) Wie können solche Metriken in eine Pipeline zum AOI-System für Qualitätskontrollaufgaben integriert und umgesetzt werden?

Zur Beantwortung dieser Forschungsfragen wird die folgende Struktur verfolgt: Abschn. 2 fasst den Hintergrund zusammen. Es werden potentiell relevante Bewertungsmetriken und spezifische Anforderungen abgeleitet, die es zu berücksichtigen gilt. Darauf basierend werden in Abschn. 3 verwandte Arbeiten diskutiert, die sich mit der Integration von Bewertungsmetriken in Qualitätskontrollaufgaben beschäftigen. In Abschn. 4 wird eine Auswahl von Bewertungsmetriken vorgestellt, die angesichts der Anforderungen potentiell geeignet sind. Die Implementierung der am besten bewerteten Metrik mit Hilfe des Demonstrators wird in Abschn. 5 beschrieben. Entsprechend werden in Abschn. 6 beobachtete Ergebnisse aus der Implementierung diskutiert.

2 Hintergrund

Im Folgenden werden die Klassen potentiell relevanter Bewertungsmetriken der Bildqualität vorgestellt. Anschließend werden spezifische Anforderungen der Bewertungsmetriken abgeleitet, die es bei der Integration in ein AOI-System zu berücksichtigen gilt.

2.1 Bildqualitätsmetriken

Nach Dosselmann et al. [4] sind Bildqualitätsmetriken (IQM) ein Thema intensiver Forschung. Die entsprechenden Algorithmen können grundsätzlich in drei Klassen eingeteilt werden: No Reference (NR)-Methoden stützen sich nicht auf ein Referenzbild zur Bestimmung einer Qualitätsbewertung. Reduced Reference (RR)-Methoden verwenden nur Teilinformationen über das Originalbild. Sie werden z. B. in Übertragungssystemen mit begrenzter Bandbreite eingesetzt. Full Reference (FR)-Methoden verwenden das komplette Originalbild als Referenz, um eine Qualitätsbewertung zu berechnen. In den letzten Jahren zeigt die Entwicklung einen Trend zu NR-Bewertungsmethoden, die keine Informationen über ein Originalbild erfordern. Diese Art von IQM eignen sich für den Einsatz in der Qualitätskontrolle, da sie dazu verwendet werden können, die aktuelle Bildqualität abzuschätzen und vorzuschlagen, ob ein bestimmtes Bild für eine Entscheidung über die Produktqualität geeignet ist oder ob äußere Faktoren eine fundierte Entscheidung verhindern könnten.

2.2 Anforderungen für den Einsatz einer IQM in der Qualitätskontrolle

Die Aussagekraft der Entscheidung, ob die Qualität eines hergestellten Produktes ausreichend ist, hängt von der Qualität der verwendeten Daten, d. h. hier der Bilder, ab. Eine schlechte Bildqualität führt zu schlechten Entscheidungen. Daher müssen die eingehenden Bilder hinsichtlich einer ausreichenden Bildqualität geprüft werden, bevor eine Entscheidung getroffen werden kann. Dies ist umso wichtiger, wenn die Führung des zu untersuchenden Objektes von Hand durchgeführt wird. Dadurch können Ungenauigkeiten bei der Bewegung und Positionierung entstehen, was wiederum unscharfe Bilder zur Folge hätte. Außerdem gibt es bei manueller Positionierung keine absolute Referenz für jedes Produkt, das geprüft wird. Dementsprechend muss das IQM ohne ein Referenzbild arbeiten (A1). Außerdem ist eine hohe Vorhersagegenauigkeit für die Bildqualität erforderlich, um zuverlässig zwischen Bildern mit ausreichender Qualität für die Aufgabe und solchen mit unzureichender Qualität unterscheiden zu können (A2). Eine weitere wesentliche Anforderung an ein IQM, welches in der Qualitätskontrolle eingesetzt werden soll, ist, dass es auch ohne Kenntnis der Fehler, welche die Bildqualität verschlechtern könnten, zuverlässig arbeiten muss (A3). Die Unschärfe eines Bildes stellt jedoch die wahrscheinlichste Ursache für eine schlechte Bildqualität bei Qualitätskontrollaufgaben dar [5], sodass ein IQM eine hohe Genauigkeit bei der Unschärfeerkennung aufweisen sollte (A4). Unschärfe kann durch Bewegung oder ungenaue Positionierung des zu prüfenden Objekts entstehen. Darüber hinaus sollte das IQM in der Lage sein, im Hinblick auf das jeweilige Anwendungssystem in Echtzeit zu arbeiten (A5). Dies kann auch von den weiteren Verarbeitungsschritten abhängen. Folglich müssen IQM in Bezug auf die Berechnung effizient sein. Schließlich wäre es von Vorteil, wenn ein IQM an verschiedene Anwendungen anpassbar wäre (A6).

3 Verwandte Arbeiten

Es existieren bereits umfangreiche Arbeiten zu verschiedenen IQM-Algorithmen und -Anwendungen. In [6] werden als typische Anwendungen des IQM die Bewertung von Algorithmen zur Bildverbesserung und -restaurierung, die Beurteilung, ob die aktuelle Bildqualität die weitere Verwendung der Bilder beeinflusst, wie z. B. in Erkennungsalgorithmen, und die Verwendung als Filter in Image Retrieval Systemen beschrieben. Die beiden letztgenannten Anwendungsfälle sind die für Qualitätskontrollsysteme relevant. Die abgerufenen Bilder müssen so gefiltert werden, dass ihre Qualität die Entscheidung über die Produktqualität nicht beeinträchtigt.

Eine beispielhafte Anwendung im Bereich der Fingerabdruckerkennung wird in [7] diskutiert. Hier stellen die Autoren fest, dass IQM für eine qualitativ hochwertige Erkennungsleistung von Fingerabdrücken unerlässlich sind. Die jeweils vorgestellten Methoden sind jedoch sehr spezifisch für Fingerabdruckbilder ausgelegt. In [8] gehen die Autoren

noch einen Schritt weiter und nutzen das IQM zur Erkennung gefälschter biometrischer Daten, um den Zugang mit gefälschten biometrischen Proben zu verhindern.

Ein weiteres Anwendungsgebiet ist die Forschung im Bereich der medizinischen Bildgebung. Bilder der Magnetresonanztomographie (MRT) müssen hinsichtlich ihrer Bildqualität untersucht werden, um valide Schlussfolgerungen ziehen zu können. In [9] wurden mehrere Bildqualitätsmetriken eingeführt, um die Qualität von MRT-Bildern objektiv und automatisch zu bewerten. Neben den MRT-Bildern wurden auch Computertomographie- (CT) und Ultraschallbilder mit IQM analysiert, um die Bildqualität objektiv und automatisch zu überprüfen [10].

In [11] betonen die Autoren die Bedeutung qualitativ hochwertiger Bilder für die robotergestützte Qualitätsprüfung von Teilen. Die Bildqualität kann jedoch unter einer unzureichenden Roboterbewegung leiden. Daher schlagen sie eine Optimierung der Robotertrajektorien auch im Hinblick auf die Qualität der entlang der Bahn aufgenommenen Bilder vor.

Lee et al. [12] stellen in ihrer Arbeit einen Ansatz zur Erkennung und Diagnose von Rissen in Brücken vor. Der Einsatz von unbemannten Luftfahrzeugen (UAV) ermöglicht dabei eine bessere Zugänglichkeit des Inspektionsbereichs. Die Qualität der mit dem UAV aufgenommenen Bilder ist jedoch sehr unbeständig. Aus diesem Grund schlagen sie eine Bewertung der Bildqualität vor der Durchführung der Risserkennung vor.

4 Auswahl und Integration einer Bewertungsmetrik für AOI in Qualitätskontrollaufgaben

Wie bisher erläutert wurde, ist es unerlässlich, Bilder von hoher Qualität zu haben, um eine valide Entscheidung darüber treffen zu können, ob ein hergestelltes Produkt von ausreichender Qualität ist oder nicht. Besonders in Bereichen mit schnell wechselnden Produkten ist es kaum möglich, Trainingsdaten mit Qualitätskennzeichnungen zu erstellen. IQM müssen folglich meinungsfrei sein, d. h. sie können sich nicht auf einen Datensatz von Bildern mit Qualitätskennzeichnungen stützen. Daher wurden aus dem Stand der Technik IQM extrahiert, die nicht meinungsabhängig sind. Die folgenden Methoden wurden als Kandidaten für ein IQM in Qualitätskontrollanwendungen identifiziert. Um den am besten geeigneten Algorithmus für die automatisierte bildverarbeitungsbasierte Qualitätskontrolle zu finden, wurden sie alle im Hinblick auf die in Abschn. 2.2 genannten Anforderungen bewertet und verglichen. Um die Leistung der Unschärfeerkennung zu bewerten, wurde der Vergleich aus [5] herangezogen. Die Ergebnisse sind in Tab. 1 veranschaulicht.

FISH [17] ist ein Schätzer für die Schärfe eines Bildes, die als Hauptursache für Bildverzerrungen bei Qualitätskontrollaufgaben identifiziert wurde. Die Methode basiert auf einer diskreten Wavelet-Transformation (DWT) und der Berechnung der Log-Energien in jedem der aus der DWT resultierenden Teilbilder. Diese Methode geht dementsprechend

Tab. 1 Bewertung der IQM-Kandidaten auf der Grundlage der Literatur, um eine Auswahl für die automatisierte bildgestützte Qualitätskontrolle zu treffen. Die grundlegenden Anforderungen wurden mit einem (+) bewertet, wenn sie erfüllt sind, und mit einem (–), wenn sie nicht erfüllt sind. Binäre Anforderungen werden mit 0 bewertet, wenn sie nicht erfüllt sind, und mit 1, wenn sie erfüllt sind. Die restlichen Anforderungen werden mit einer Punktzahl von 0 für überhaupt nicht erfüllt bis 3 für vollständig erfüllt bewertet

IQM	A1: NR	A2: Qualitätsbewertung	A3: Fehlertypen	A4: Unschärfeerkennung	A5: Echtzeitfähigkeit	A6: Anpassungsfähigkeit	Ergebnis
BRISQUE [13]	+	+	+	1	3	0	4
NIQE [14]	+	+	+	1	3	1	5
IL-NIQE [15]	+	+	+	3	0	1	4
PIQUE [16]	+	+	+	1	2	0	3
FISH [17]	+	+	–	/	/	/	/

von einer auftretenden Verzerrung in Form von Unschärfe aus und ist nicht an andere Typen anpassbar.

BRISQUE [13] basiert auf einem Support Vector Regression-Modell, das anhand eines Datensatzes von Bildern mit Qualitätskennzeichnungen trainiert wird, die von menschlichen Bewertungen abgeleitet sind. Das Ergebnis ist ein numerischer Index. Die Methode selbst kennt die erwartete Art der Bildverzerrung nicht. Die tatsächlich auftretenden Arten sollten jedoch Teil der Trainingsdaten sein, um sie genau zu bewerten. Der Rechenaufwand ist nach Angaben der Autoren so gering, dass das Verfahren in Echtzeit ausgeführt werden kann. Durch die Verwendung von Trainingsdaten, die von Menschen bewertet werden, ist es nicht meinungsunabhängig und kaum an neue Fälle anpassbar, die nicht von den Trainingsdaten abgedeckt werden.

NIQE [14] leitet einen Qualitätsindex aus einem Modell ab, das im Voraus anhand eines Datensatzes unverzerrter Bilder bestimmt wird. Aus diesem Datensatz wird ein Modell konstruiert, das auf speziell ausgewählten Merkmalen basiert. Der Bildqualitätsindex ist dann der Abstand des Qualitätsmodells, wie es auf ein Testbild angewendet wird, von dem vorgegebenen Modell. Dieses IQM ist also verzerrungs- und meinungsunabhängig. Da der Trainingsdatensatz aus unverzerrten Bildern besteht, lässt er sich leicht an neue Anwendungsfälle anpassen.

IL-NIQE [15] ist im Wesentlichen eine Weiterentwicklung des oben erwähnten NIQE [14]. Diese führt mehrere zusätzliche Merkmale ein und berechnet lokale Indizes auf Blockebene anstelle eines globalen Index. Auch dieser Algorithmus ist nicht meinungs- und verzerrungsabhängig. Er steigert die Unschärfeerkennungsleistung von NIQE [14] erheblich. Aufgrund der erhöhten Komplexität ist es nicht mehr möglich, dieses IQM in Echtzeit auszuführen. Es ist jedoch genauso anpassungsfähig wie NIQE [14], da es auch mit einem Datensatz unverzerrter Bilder arbeitet.

PIQUE [16] berechnet die Qualität ebenfalls auf Blockebene. Es sucht nach räumlich aktiven Blöcken, die für die menschliche Bewertung der Bildqualität wesentlich sind.

Jeder räumlich aktive Block wird auf auffällige Verzerrungen und Rauschen untersucht, woraus eine Punktzahl für jeden Bildblock abgeleitet wird. Die Bewertung der Bildqualität ist dann die Summe aller Blockbewertungen. Somit ist dieses IQM auch verzerrungs- und meinungsunabhängig. Es erfordert keine Trainingsdaten und ist folglich übertragbar. Es wurde jedoch noch nie an Bildern aus technischen Umgebungen evaluiert, so dass unklar bleibt, ob es auch für dieses Szenario geeignet ist.

5 Versuchsaufbau

Um die Eignung der Bildmetrik zur Bewertung der Bildqualität sowie die Gewährleistung der Leitungserhaltung des Bildauswertungsprozesses durch die Integration der Bildmetrik in Form eines Bildbewertungsmoduls zu evaluieren, werden Bilddaten zweier verschiedener Anwendungsfälle künstlich verzerrt. Daraufhin wird NIQE als die am besten geeignete Bildmetrik zur Integration in ein AOI-System hinsichtlich ihrer Leistungsfähigkeit zur Bewertung der Bildqualität evaluiert. Abschließend wird die Auswirkung der Bildverzerrung auf die Leistung des Bildauswertungsprozesses sowie der Nutzen der Integration eines Bildbewertungsmoduls in ein AOI-System bewertet.

5.1 Datensätze

Zur Evaluation werden zwei verschiedene Anwendungsfälle betrachtet. Der erste Anwendungsfall ist die Inspektion von Metalloberflächen mit dem Ziel der Klassifikation von verschiedenen Defekttypen. Zu diesem Zweck wird der von der Northeastern University veröffentlichte NEU-DET-Datensatz [18] verwendet. Dieser Datensatz umfasst 1800 Bilder mit sechs typischen Defekten auf Metalloberflächen. Jede der sechs Defektkategorien umfasst 300 Bilder.

Der zweite Anwendungsfall ist die Inspektion von Leiterplatten. Dabei gilt es, die auf der Leiterplatte vorliegenden Defekte nicht nur zu klassifizieren, sondern auch zu lokalisieren. Dazu wird die Erweiterung [19] des von der Open Lab on Human Robot Interaction der Peking University veröffentlichten PCB-Datensatzes verwendet. Dieser Datensatz umfasst 12.428 Bilder mit sechs Arten von Defekten auf Leiterplatten.

5.2 Verzerrungsarten und -stufen

Für beide Anwendungsfälle werden für die im entsprechenden Datensatz vorhandenen Bilder eine Vielzahl zusätzlicher Bilder mit unterschiedlichen Bildverzerrungen erzeugt. Es werden vier Arten üblicher Verzerrungen berücksichtigt: Tiefenunschärfe, Bewegungsunschärfe, Rauschen und Belichtung. Jede Verzerrung wird separat betrachtet. Tiefenunschärfe kann auftreten, wenn die Kamera nicht richtig auf das zu verarbeitende Objekt

fokussiert ist. Für diese Art von Unschärfe wird ein Gauß-Kernel verwendet. Die Standardabweichung σ des Gauß-Kernels wird in Schritten von 0,1 zwischen 0,1 und 10 variiert. Neben der Tiefenunschärfe ist die Bewegungsunschärfe eine weitere Ursache für eine schlechte Bildqualität. Zu diesem Zweck wird eine Bewegung entlang der Horizontalen simuliert, indem die Pixel des Originalbildes auf eine bestimmte Anzahl von Pixeln im verzerrten Bild gestreckt werden. Die Anzahl der Pixel variiert von 1 bis 50 Pixel mit einer Schrittweite von 0,5 Pixel. Rauschen kann durch die Verwendung von Kamerasensoren minderer Qualität verursacht werden. Einzelne Pixel im Sensor können defekt sein und unabhängig von der Szene ein konstantes Signal liefern. Dieses Rauschen kann als Impulsrauschen modelliert werden. Die Wahrscheinlichkeit, dass ein Pixel vom Rauschen betroffen ist, wird von 0 bis 10 Prozent mit einer Schrittweite von 0,1 Prozent variiert. Schließlich wird die Auswirkung von künstlicher Aufhellung und Abdunklung auf das Bild bewertet, um reale Belichtungsfehler zu simulieren. Zu diesem Zweck wird zu jedem Helligkeitswert pro Kanal ein bestimmter Wert addiert. Dieser Wert wird von -150 bis $+150$ mit einer Schrittweite von 4 variiert. Die Unter- und Obergrenze der Helligkeitswerte sind auf 0 und 255 begrenzt.

5.3 Bildauswertungsverfahren

Die Leistung der Integration des Bildbewertungsmoduls in ein AOI-System wird für zwei Arten von Inspektionsaufgaben evaluiert. Für die Oberflächeninspektion des NEU-DET Datensatzes werden vier repräsentative neuronale Netze zur Klassifikation von verschiedenen Defekttypen betrachtet. Bei den in dieser Arbeit getesteten neuronalen Netze handelt es sich um AlexNet [20], VVG16 [21], ResNet18 [22] und DenseNet161 [23]. Diese Netze wurden alle auf dem ImageNet-Datensatz [24] vortrainiert. Die Implementierung der Netze erfolgte im Framework PyTorch [25] und verwendete die vortrainierten Modellgewichte der TorchVision Bibliothek. Um eine hohe Genauigkeit für die Klassifikation der verschiedenen Defekttypen zu gewährleisten, werden die betrachteten neuronalen Netze mit Anfangsgewichten aus den vortrainierten Modellen zur Feinabstimmung neu trainiert. Dazu wurden die Modellgewichte aller Schichten der neuronalen Netze an den Anwendungsfall angepasst. Während des Trainings wurden die neuronalen Netze hinsichtlich der Kreuzentropie als Verlustfunktion über 15 Epochen mit Hilfe des SDG-Optimierers mit einer schrittweise sinkenden Lernrate beginnend bei 0,01 und einer Verringerung um den Faktor 0,1 alle 5 Epochen optimiert. Die Visualisierung der Verlust- und Genauigkeitskurven für die Trainings- und Validierungsdaten zeigt eine Konvergenz und damit eine entsprechende Fähigkeit zur Generalisierung.

Zur Evaluation der Leistungserhaltung des Bildauswertungsprozesses durch den Einsatz des Bildbewertungsmoduls wurde neben der Klassifikation ebenfalls die Auswirkung des Bildbewertungsmoduls auf die Leistung eines Objekterkennungsverfahrens bewertet. Dazu wurde das YOLOv5-Modell [26] mit vortrainierten Modellgewichten verwendet. Dessen Implementierung erfolgte ebenfalls in PyTorch [25]. Das Modell wurde auf dem

COCO-Datensatz [27] vortrainiert und auf den in Abschn. 5.1 beschriebenem Leiterplatten-Datensatz feinabgestimmt. Das Training erfolgte über 60 Epochen mit einer Batchgröße von 16. Die restlichen Hyperparameter blieben im Vergleich zu Jocher et al. [26] unverändert.

6 Auswertung und Diskussion

Die für die Integration in ein AOI-System am besten geeignete Bildmetrik, NIQE, wird zuerst hinsichtlich ihrer Leistungsfähigkeit zur Erkennung der vorgestellten Verzerrungsarten und -stufen bewertet. Anschließend wird die Bildmetrik modularisiert und in ein AOI-System integriert. Das zusätzlich integrierte Modul zur Bildbewertung dient der Entscheidung, ob ein Bild weiterverarbeitet werden kann oder als ungeeignet ausscheidet. Diesbezüglich wird die Gewährleistung qualitativ hochwertiger Eingabedaten sowie die dadurch ermöglichte Leistungserhaltung des Bildauswertungsprozess bewertet.

6.1 Bewertung der Leistungsfähigkeit zur Bildqualitätserkennung

Zur Bewertung der Leistungsfähigkeit der Erkennung der Bildqualitätsdegradation durch die vorgestellten Verzerrungsarten und -stufen werden neben dem NIQE-Modell mit Standardeigenschaftswerten, die von der in [14] genannten Bilddatenbank abgeleitet sind, eine zweite an die entsprechenden Anwendungsfälle angepasste Version erzeugt. Für die Erstellung der anwendungsfallspezifischen NIQE-Modelle werden für beide Anwendungsfälle Reihen von Bildern als Referenz in einer Bilddatenbank zur Verfügung gestellt, aus denen die anwendungsfallspezifischen Parameter für die Modelle ermittelt werden. Für die Bewertung der Funktionsüberprüfung mittels künstlich verfälschter Bilder werden die von R. Dosselmann et al. [1] genannten Kriterien Vorhersagegenauigkeit, Monotonie und Konsistenz der Bewertung überprüft. Die Vorhersagegenauigkeit wird mithilfe des linearen Korrelationskoeffizienten von Pearson (PLCC) zwischen der NIQE-Bewertung und der Wurzel der mittleren Fehlerquadratsumme (RMSE) der verfälschten Bilder im Vergleich zu dem Originalbild bestimmt. Für die Monotonie der Vorhersage wird der Spearman’s Rangordnungs-Korrelationskoeffizient (SROCC) zwischen selbigen verwendet.

Abb. 1 zeigt die Ergebnisse der Experimente bezüglich der Verzerrungsarten und -stufen.

Tiefenunschärfe. Es ist ein erheblicher Unterschied des Verlaufes der Qualitätsbewertung zwischen dem NIQE-Modell mit Standardeigenschaftswerten und dem mit anwendungsfallspezifischen Werten zu erkennen. Das NIQE-Modell mit Standardeigenschaftswerten zeigt bei geringer Erhöhung der Standardabweichung des Gauß’schen Filters eine Verbesserung der Bildqualität. Anschließend steigt die Bewertungsmetrik mit zunehmender Standabweichung leicht über die Bewertung des Originalbildes, bleibt jedoch nahezu konstant. Dies weist darauf hin, dass sich das NIQE-Modell mit Standardeigenschafts-

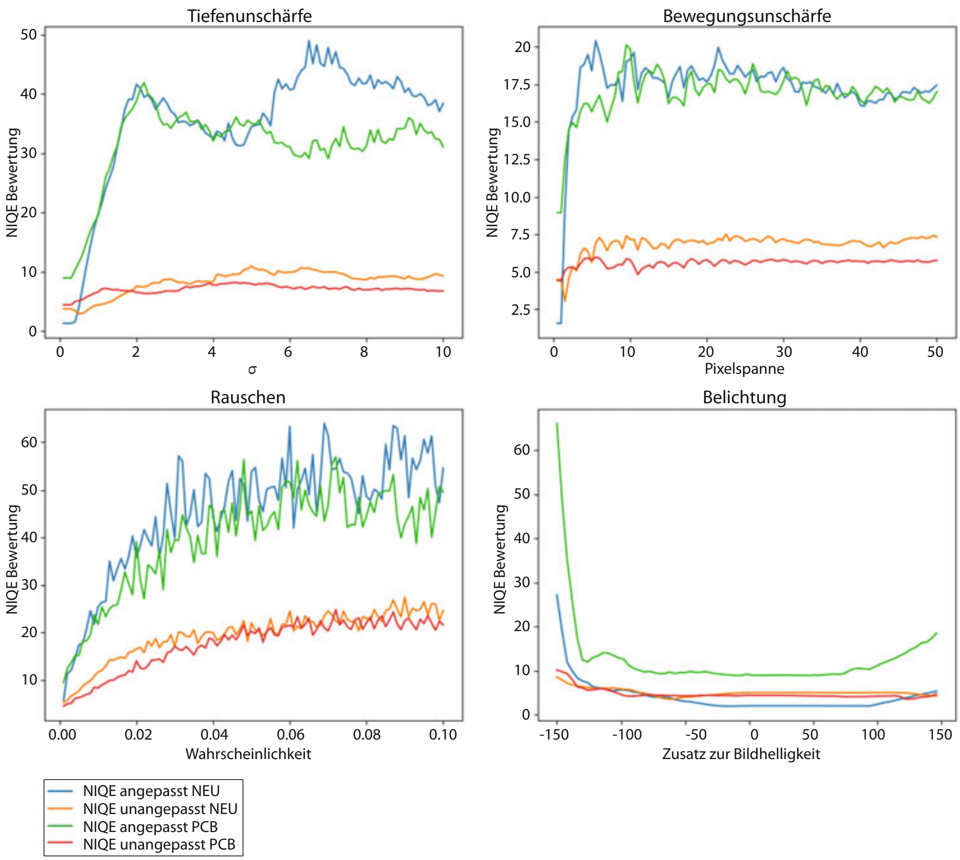


Abb. 1 Bildqualitätsbewertung in Abhängigkeit der verschiedenen Verzerrungsarten und -stufen

werten nicht für die Bewertung einer durch einen Gauß'schen Filter erzeugten Unschärfe eignet. Die geringe Vorhersagegenauigkeit und Monotonie zeigen sich ebenfalls in geringen PLCC- und SROCC-Werten. Bei den anwendungsfallspezifischen NIQE-Modellen ist hingegen ein klarer Unterschied zwischen der Bewertung des Originalbildes und der verfälschten Bilder sichtbar. Es ist eine hohe Leistungsfähigkeit zur Erkennung der Qualitätsabnahme bei geringer Verzerrung zu erkennen. Bei hoher Verzerrung hingegen wird zwar eine geringe Bildqualität erkannt, jedoch wird nicht der Grad der Qualitätsabnahme angezeigt. Die hohen PLCC- und SROCC-Werte bestätigen die hohe Vorhersagegenauigkeit und Monotonie der Modelle.

Bewegungsunschärfe. Analog zur Tiefenunschärfe ist auch bei der Erkennung der Bewegungsschärfe ein Unterschied zwischen der Leistung der NIQE-Modelle mit Standardeigenschaftswerten und denen mit anwendungsfallspezifischen Werten zu erkennen. Dabei ist der Verlauf der Qualitätsbewertung nahezu identisch mit dem der Tiefenunschärfe. Die PLCC- und SROCC-Werte der anwendungsfallspezifischen NIQE-Modelle

verweisen auf eine mittlere Korrelation hin. Dies ist dadurch begründet, dass zwar eine geringe Bildqualität erkannt wird, die Qualitätsverschlechterung im Bereich niedriger Qualität jedoch nicht ermittelt werden kann.

Die anwendungsfallspezifischen NIQE-Modelle sind somit in der Lage, ein Bild ohne Bewegungsunschärfe von einem Bild mit künstlicher Bewegungsunschärfe zu unterscheiden. Die Genauigkeit der Qualitätsbewertung lässt allerdings bei zunehmender Intensität des Fehlers nach.

Rauschen. Die Verläufe der NIQE-Modelle mit Standardeigenschaftswerten und der mit anwendungsfallspezifischen Werten zur Qualitätsbewertung von Bildern mit künstlich hinzugefügtem Rauschen weisen starke Ähnlichkeiten auf. Dabei ist ein Anstieg der Qualitätsbewertungskennzahl relativ zur Qualitätsdegradation zu erkennen. Allerdings unterscheiden sich die Bewertungen zwischen den NIQE-Modellen stark in ihrer Skalierung. Die Maximalbewertung der Modelle mit Standardeigenschaftswerten liegt dabei weit unter der von Modellen mit anwendungsfallspezifischen NIQE. Die PLCC- und SROCC-Werte sind in beiden Fällen hoch und nahezu identisch. Dies zeigt, dass NIQE geeignet ist, künstliche hinzugefügtes Bildrauschen präzise zu bewerten.

Belichtung. Es existieren starke Unterschiede im Verlauf und den Maximalwerten zwischen der Bewertung der NIQE-Modelle mit Standardeigenschaftswerten und der Bewertung der Modelle mit anwendungsfallspezifischen Werten. Die Bewertungen der NIQE-Modelle mit Standardeigenschaftswerten bleiben annähernd konstant auf einem Niveau und sind somit nicht zu Erkennung und Bewertung künstlich hinzugefügter Belichtungsunterschiede geeignet. Die anwendungsfallspezifischen NIQE-Modelle hingegen zeigen Qualitätsverluste durch Belichtungsunterschiede an. Bei moderaten Belichtungsunterschieden ist der Anstieg der Bewertung jedoch nur sehr gering. Erst mit zunehmender Verdunkelung oder Aufhellung der Bilder werden deutliche Qualitätsverluste ermittelt. Die gleichen Erkenntnisse zeigen sich ebenfalls in den PLCC- und SROCC-Werten. Zusätzlich ist zu erkennen, dass Qualitätsverluste durch Verdunklung der Bilder besser erkannt werden als durch Aufhellung. Insgesamt ist NIQE somit geeignet, künstliche hinzugefügtes Belichtungsunterschiede zu erkennen. Es ist jedoch unpräzise bei geringen Belichtungsunterschieden.

Konsistenz der Vorhersage. Die hohe Genauigkeit als auch die hohe Monotonie der Vorhersagen verschiedener Bilder zweier Anwendungsfälle mit unterschiedlichen Verzerrungsarten und -stufen zeigt, dass eine Konsistenz der Vorhersage bei NIQE-Modellen mit anwendungsfallspezifischen Werten vorherrscht. Zudem zeigte das frühzeitige Erreichen eines Plateaus der Bewertung künstlich hinzugefügter Tiefen- und Bewegungsunschärfe, dass eine geringe Bildqualität klar erkennbar ist, die Qualitätsabnahme bei Bildern mit sehr niedriger Qualität jedoch nicht mehr erkannt wird.

6.2 Bewertung der Leistungserhaltung des Bildauswertungsprozess

Um den Nutzen des Einsatzes eines Bildbewertungsmoduls zu bestimmen, wurde die Leistung der in Abschn. 5.2 genannten Bildauswertungsverfahren zur Klassifikation und Objekterkennung auf künstlich verzerrten Bildern der beschriebenen Arten und Stufen der Bildverzerrung evaluiert.

Klassifikation. Abb. 2 zeigt die Klassifikationsgenauigkeit der neuronalen Netze in Bezug auf die verschiedenen Bildqualitätsverzerrungen. Alle Netze sind empfindlich gegenüber Unschärfe. Bei geringer Unschärfe ist jedoch zunächst kein Leistungsabfall der Klassifizierungsgenauigkeit zu erkennen. Bei mittleren Unschärfegraden nimmt die Klassifizierungsgenauigkeit der Netze allerdings deutlich ab. Diese Verringerung der Leistung lässt sich dadurch erklären, dass die Unschärfe die Texturen in den Bildern anhand derer die Netze das Bild klassifizieren, entfernt wird. Diese Erkenntnisse gelten sowohl für die Tiefenunschärfe als auch für die Bewegungsunschärfe. Die Klassifikationsgenauigkeit der Netze von Bildern mit Rauschen weist eine ähnliche Leistungsabnahme auf. Anders als bei der Unschärfe führt hier jedoch bereits ein sehr geringes Rauschen zu deutlich

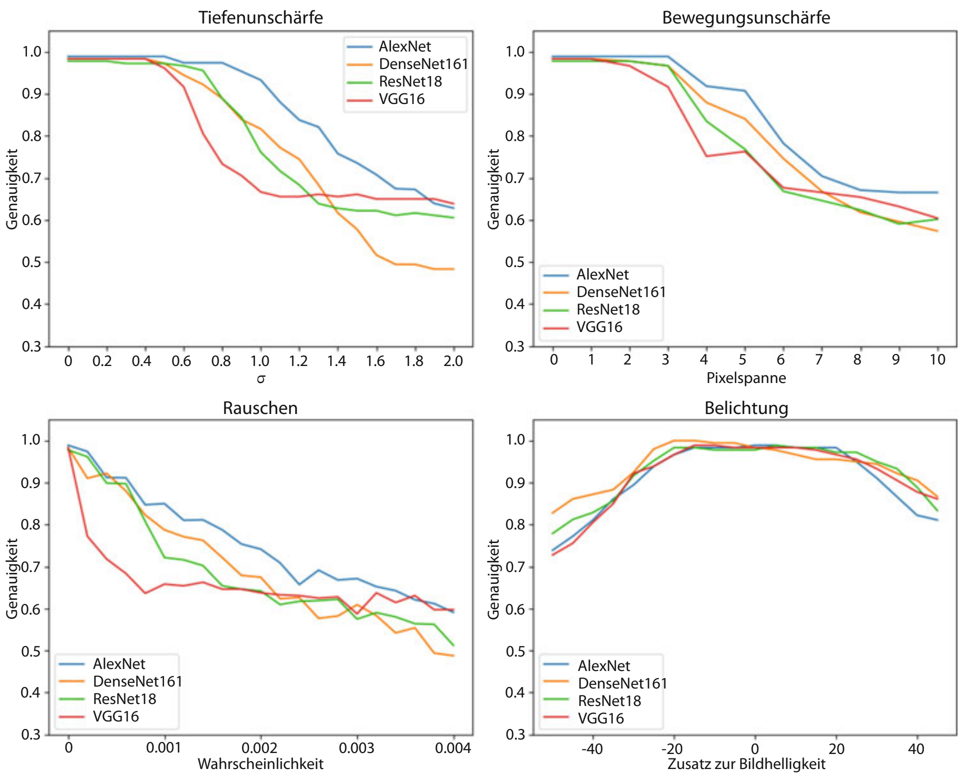


Abb. 2 Leistungsbewertung der neuronalen Netze in Abhängigkeit der verschiedenen Verzerrungsarten und -stufen

geringerer Genauigkeit der Netze. Die Visualisierung der Pixel-Attribution durch die Verwendung von Salienzkarten zeigt, dass die für die Klassifikation relevanten Pixel sehr gering, jedoch über den Großteil des Bildes verteilt sind. Dies könnte die Ursache für den deutlichen Leistungsabfall bei bereits geringem Rauschen sein. Weniger empfindlich sind die Netze gegenüber Aufhellung der Bilder. Die Abdunkelung der Bilder hingegen weist eine stärke Abnahme der Klassifikationsgenauigkeit auf.

Abb. 3 zeigt die Klassifikationsgenauigkeit der neuronalen Netze mit vorgeschaltetem Bildbewertungsmodul in Bezug auf die verschiedenen Bildqualitätsverzerrungen. Die Integration des Bildbewertungsmoduls in das AOI-System ermöglicht die Leistungserhaltung des Bildauswertungsprozess in Form der neuronalen Netze. Das Bildbewertungsmodul bewirkt das Herausfiltern und Entfernen von Bildern mit geringer Qualität. Trotz steigender Qualitätsverzerrung bleibt die Klassifikationsgenauigkeit der neuronalen Netze konstant, wobei die Anzahl der herausgefilterten Bilder mit zunehmender Qualitätsverzerrung steigt. Den neuronalen Netzen werden ausschließlich Bilder ausreichender Qualität zugeführt. Dies ermöglicht den Erhalt der Klassifikationsgenauigkeiten.

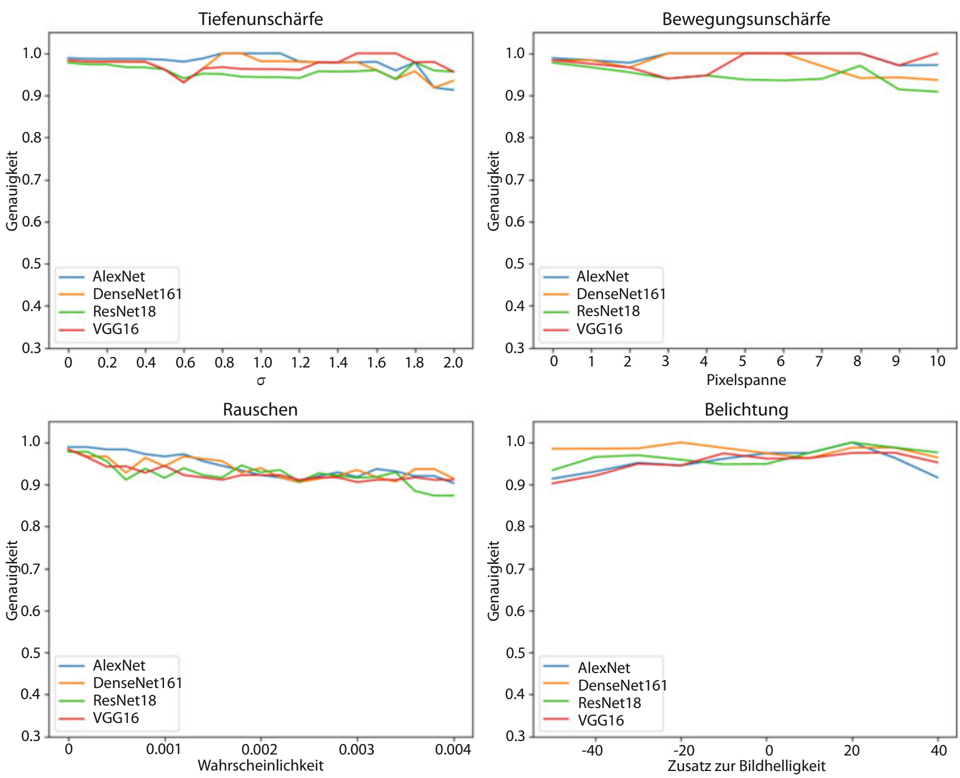


Abb. 3 Leistungsbewertung der neuronalen Netze mit vorgeschaltetem Bildbewertungsmodul in Abhängigkeit der verschiedenen Verzerrungsarten und -stufen

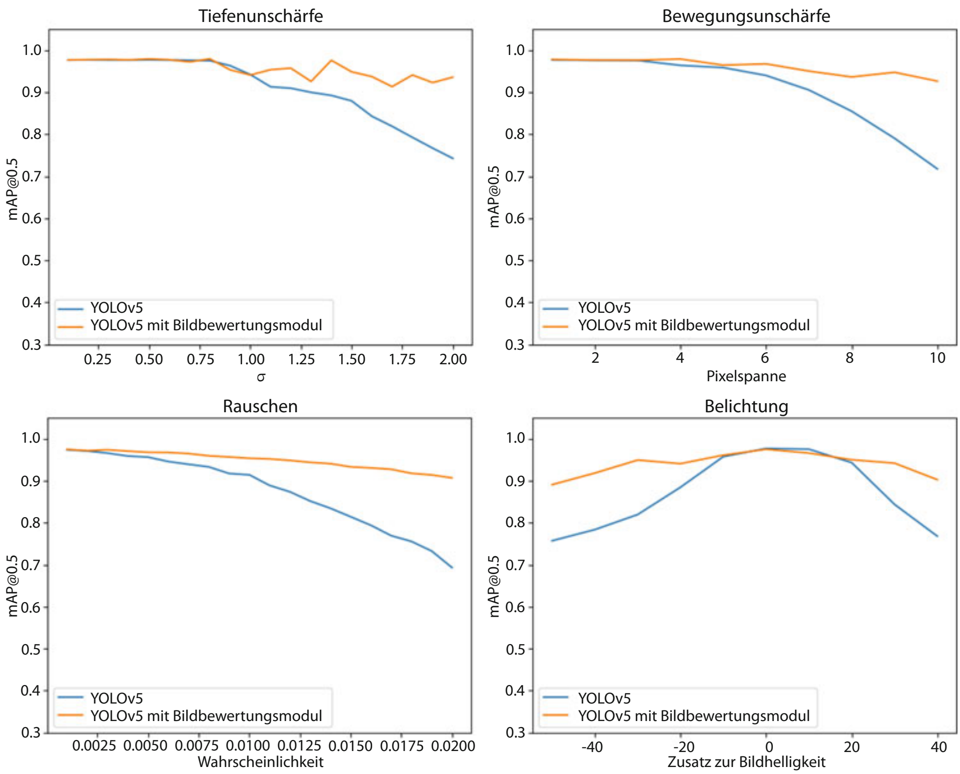


Abb. 4 Leistungsbewertung des Objekterkennungssystems ohne und mit vorgeschaltetem Bildbewertungsmodul in Abhängigkeit der verschiedenen Verzerrungsarten und -stufen

Objekterkennung. Abb. 4 zeigt die für die Leistungsbewertung von Objekterkennungssystemen geeignete Metrik mAP@0,5 des trainierten YOLOv5-Modells in Bezug auf die verschiedenen Bildqualitätsverzerrungen. Dabei sind die Erkenntnisse analog zu den Klassifikationsmodellen. Das Objekterkennungsmodell ist jedoch deutlich widerstandsfähiger gegenüber rauschbehafteten Bildern.

Auch bei der Objekterkennung wird der Nutzen der Integration des Bildbewertungsmoduls in das AOI-System ersichtlich. Die Leistungserhaltung des Bildauswertungsprozess kann trotz zunehmender Qualitätsverzerrung gewährleistet werden.

7 Diskussion und Schlussfolgerung

Die Ergebnisse zeigen, dass sich NIQE-Modelle, die an die Bilddaten des jeweiligen Anwendungsfalls angepasst sind, zur Erkennung und Bewertung der beschriebenen Verzerrungsarten und -stufen eignen. Dies gilt vor allem für geringe Verzerrungsstufen. Mit zunehmender Verzerrung wird weiterhin eine geringe Qualität ermittelt, jedoch wird die Bewertung des Qualitätsverlustes weniger präzise.

Die Experimente weisen zudem darauf hin, dass die getesteten neuronalen Netze gegenüber den vorgestellten Bildverzerrungen anfällig sind. Dies gilt sowohl für die Klassifikation als auch für die Objekterkennung. Die Integration eines Bildbewertungsmoduls in ein AOI-System zum Herausfiltern von qualitativ minderwertigen Bildern ermöglicht die Leistungserhaltung des jeweiligen Bildauswertungsprozesses.

Da die herausgefilterten Bilder aus dem AOI-System entfernt werden, ist die Integration eines Bildbewertungsmoduls vor allem dann geeignet, wenn von dem zu untersuchenden Objekt eine Vielzahl von Bildern, beispielsweise in Form eines Bilderstroms, aufgenommen werden.

Es kann sinnvoll sein, die neuronalen Netze mit Bildern niedriger Qualität zu trainieren, um die Anzahl der für die Entscheidung des Bildauswertungsprozesses geeigneten Bilder zu erhöhen. Dabei ist zu beachten, dass das Training mit Bildern niedriger Qualität die Leistung der neuronalen Netze bei Bildern niedriger Qualität verbessern, gleichzeitig aber zu einer geringeren Leistung bei Bildern mit hoher Qualität führen kann. Diese Entscheidung sollte daher anwendungsfallspezifisch getroffen werden.

Ein mögliches Hindernis stellt das Herausfiltern von Bildern mit niedriger Qualität in Bildabschnitten, die für die Entscheidung der Bildauswertung nicht relevant sind, dar. In zukünftigen Arbeiten planen wir, entscheidungsrelevante Bildinformation beispielsweise in Form von Pixel-Attribution der neuronalen Netze in die Bildqualitätsbewertung einzubeziehen und dadurch die Auswahl der für den Bildauswertungsprozess geeigneten Bilder weiter zu verbessern.

Danksagung Die Autoren bedanken sich für die Förderung bei dtec.bw – Zentrum für Digitalisierungs – und Technologieforschung der Bundeswehr – Projekte EKI und ProMoDi.

Literatur

1. Korodi A, Anitei D, Boitor A, Silea I (2020) Image-processing-based low-cost fault detection solution for end-of-line ECus in automotive manufacturing. Sensors 20:3520. https://doi.org/10.3390/s20123520
2. Li S, Zheng P, Zheng L (2021) An AR-assisted deep learning-based approach for automatic inspection of aviation connectors. IEEE Trans Ind Inf 17:1721–1731. https://doi.org/10.1109/TII.2020.3000870

3. Dodge S, Karam L (2016) Understanding how image quality affects deep neural networks. In: QoMEX 2016. QoMEX 2016. 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), Lisbon, 6–8 June 2016 IEEE, Piscataway, S 1–6
4. Li L, Xia W, Lin W, Fang Y, Wang S (2017) No-reference and robust image sharpness evaluation based on multiscale spatial and spectral features. IEEE Trans Multimedia 19:1030–1040. https://doi.org/10.1109/TMM.2016.2640762
5. Bahnemiri SG, Ponomarenko M, Egiazarian K (2020) On verification of blur and sharpness metrics for no-reference image visual quality assessment. In: IEEE 22nd International Workshop on Multimedia Signal Processing. Virtually Tampere, September 21–24 IEEE, Piscataway, S 1–6 https://doi.org/10.1109/MMSP48831.2020.9287110
6. Pedersen M, Hardeberg JY (2012) Full-reference image quality metrics: classification and evaluation. Fnt Comput Graph Vis 7:1–80. https://doi.org/10.1561/0600000037
7. Alonso-Fernandez F, Fierrez-Aguilar J, Ortega-Garcia J (2022) A review of schemes for fingerprint image quality computation
8. Galbally J, Marcel S, Fierrez J (2014) Image quality assessment for fake biometric detection: application to Iris, fingerprint, and face recognition. IEEE Trans Image Process 23:710–724. https://doi.org/10.1109/TIP.2013.2292332
9. Woodard JP, Carley-Spencer MP (2006) No-reference image quality metrics for structural MRI. NI 4:243–262. https://doi.org/10.1385/NI:4:3:243
10. Chow LS, Paramesran R (2016) Review of medical image quality assessment. Biomed Signal Process Control 27:145–154. https://doi.org/10.1016/j.bspc.2016.02.006
11. Lončarević Z, Gams A, Reberšek S, Nemeca B, Škrabarc J, Skvarčd J, Ude A (2021) Specifying and optimizing robotic motion for visual quality inspection. Robot Comput Integr Manuf 72:102200. https://doi.org/10.1016/j.rcim.2021.102200
12. Lee J-H, Yoon S-S, Kim I-H, Jung HL (2018) Study on image quality assessment and processing, damage diagnosis of crack for bridge inspection based on unmanned aerial vehicle. In: ACEM18/Structures18, S 1–6
13. Mittal A, Moorthy AK, Bovik AC (2012) No-reference image quality assessment in the spatial domain. IEEE Trans Image Process 21:4695–4708. https://doi.org/10.1109/TIP.2012.2214050
14. Mittal A, Soundararajan R, Bovik AC (2013) Making a "completely blind" image quality analyzer. IEEE Signal Process Lett 20:209–212. https://doi.org/10.1109/LSP.2012.2227726
15. Zhang L, Zhang L, Bovik AC (2015) A feature-enriched completely blind image quality evaluator. IEEE Trans Image Process 24:2579–2591. https://doi.org/10.1109/TIP.2015.2426416
16. Venkatanath N, Praneeth D, Maruthi Chandrasekhar B, Channappayya SS, Medasani SS (2015) Blind image quality evaluation using perception based features. In: 2015 Twenty First National Conference on Communications (NCC 2015) Mumbai, 27 February – 1 March 2015 IEEE, Piscataway, S 1–6 https://doi.org/10.1109/NCC.2015.7084843
17. Vu PV, Chandler DM (2012) A fast wavelet-based algorithm for global and local image sharpness estimation. IEEE Signal Process Lett 19:423–426. https://doi.org/10.1109/LSP.2012.2199980
18. Song K, Yan Y (2013) A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Appl Surf Sci 285:858–864. https://doi.org/10.1016/j.apsusc.2013.09.002
19. Ding R, Dai L, Li G, Liu H (2019) TDD-net: a tiny defect detection network for printed circuit boards. CAAI Trans Intell Technol 4:110–116. https://doi.org/10.1049/trit.2019.0019
20. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60:84–90. https://doi.org/10.1145/3065386
21. Simonyan K, Zisserman A (2014) Very deep Convolutional networks for large-scale image recognition https://doi.org/10.48550/arxiv.1409.1556

22. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) https://doi.org/10.1109/cvpr.2016.90
23. Huang G, van der Maaten LZ, Weinberger KQ (2016) Densely connected convolutional networks https://doi.org/10.48550/arXiv.1608.06993
24. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115:211–252. https://doi.org/10.1007/s11263-015-0816-y
25. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala J (2019) Pytorch: an imperative style, high-performance deep learning library. Adv Neural Inf Proccess Syst 32:8026–8037
26. Jocher G, Chaurasia A, Stoken A (2022) Ultralytics/yolov5: v6.2 – YOLOv5 classification models, Apple M1, reproducibility, ClearML and Deci.ai integrations
27. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick LC (2014) Microsoft COCO: Common Objects in Context. In: Fleet D, Pajdla T, Schiele B et al (Hrsg) Computer vision – ECCV 2014 13th European conference, Zurich, September 6–12, 2014. proceedings, part V, Bd. 8693. Springer, Cham, S 740–755

DSGVO-konforme Personendetektion in 3D-LiDAR-Daten mittels Deep Learning Verfahren

Dennis Sprute, Tim Westerhold, Florian Hufen, Holger Flatt und Florian Gellert

Zusammenfassung

Im Fabrikkontext spielt die Detektion von Personen eine wichtige Rolle bei Maßnahmen zur Erhöhung der Sicherheit von Arbeitern oder zur Optimierung des Fabriklayouts. Kamerasensoren bilden die Grundlage für robuste bildbasierte Personendetektionsverfahren, werden aber aufgrund von Datenschutzaspekten häufig kritisch gesehen. Diese Bedenken können durch die Lokalisation von IoT-Devices, die von Personen getragen werden, adressiert werden, jedoch muss eine Person stets mit einem entsprechendem IoT-Device ausgerüstet sein. In diesem Beitrag wird ein alternativer Ansatz zur Adressierung der Problematik vorgeschlagen, der auf etablierten Bildverarbeitungsverfahren beruht, jedoch inhärent DSGVO-konform ist. Hierzu wird distanzmessende 3D-LiDAR-Sensorik genutzt, um 3D-Punktwolken der Umgebung aufzunehmen. Diese ermöglichen eine Detektion von Personen (Klassifikation und Lokalisation), jedoch keine Identifizierung der Personen. Hierfür wird ein Verfahren vorgestellt, das einzelne Objekte in einer Punktwolke zunächst in ein Tiefenbild umwandelt, um auf diesem anschließend robuste Bildverarbeitungsverfahren basierend

D. Sprute (✉) · T. Westerhold · F. Hufen · H. Flatt · F. Gellert
Fraunhofer IOSB, Institutsteil für industrielle Automation (IOSB-INA), Lemgo, Deutschland
E-Mail: dennis.sprute@iosb-ina.fraunhofer.de

T. Westerhold
E-Mail: tim.westerhold@iosb-ina.fraunhofer.de

F. Hufen
E-Mail: florian.hufen@iosb-ina.fraunhofer.de

H. Flatt
E-Mail: holger.flatt@iosb-ina.fraunhofer.de

F. Gellert
E-Mail: florian.gellert@iosb-ina.fraunhofer.de

V. Lohweg (Hrsg.), *Bildverarbeitung in der Automation*, Technologien für die intelligente Automation 17, https://doi.org/10.1007/978-3-662-66769-9_3

auf Deep Learning einzusetzen. Die Evaluation des Verfahrens zeigt eine Genauigkeit (Accuracy) von 98%, um zwischen Personen und anderen Objekten zu unterscheiden, und ist somit für darauf aufbauende Anwendungen gut geeignet.

Schlüsselwörter

Personendetektion · Deep Learning · 3D-LiDAR · 3D-Punktwolke · DSGVO-Konformität

1 Motivation

Die Erfassung und genaue Lokalisation von Personen im Fabrikkontext kann einen wichtigen Beitrag zur Erhöhung der Sicherheit von Arbeitern und zur Optimierung des Fabriklayouts leisten. Beispielsweise kann durch die Detektion von unautorisierten Personen in Sicherheitsbereichen oder die Erkennung von Notfallsituationen die Sicherheit erhöht werden. Zudem kann die Analyse von Personenbewegungen oder die Erkennung von Gruppenbildungen am Fließband zur Optimierung des Fabriklayouts genutzt werden. Bei der Erfassung von Personen spielt insbesondere in Deutschland und der EU der Datenschutz eine wichtige Rolle, bei dem es um den Schutz von personenbezogenen Daten von Personen geht[1]. Es sollte also nicht möglich sein, eine erfasste und lokalisierte Person zu *identifizieren*, sodass z. B. erzeugte Bewegungsprofile einer konkreten Person zugeordnet werden können. Kamerasensoren, die an geeigneten Stellen in einer Fabrik installiert sind, können sehr gut mittels heutigen Verfahren des Stands der Technik für die Detektion von Personen in Bildern genutzt werden [1, 2]. Jedoch erlauben die Bilder bei hohen Auflösungen durch die Bestimmung von Merkmalen wie Gesicht, Hautfarbe u. ä. oftmals auch eine Identifikation der entsprechenden Personen, sodass dieser Ansatz kritisch zu sehen ist. Weitere Ansätze für diese Problemstellung umfassen Systeme zur Indoor-Lokalisierung, die Personen mittels funkbasierten IoT-Devices lokalisieren können [3]. Hierbei muss jedoch jeder Arbeiter stets mit einem entsprechenden IoT-Tag ausgerüstet sein.

In diesem Beitrag wird ein alternativer Ansatz vorgeschlagen, der die Vorteile kamerabasierter Sensoren und bildbasierter Verarbeitung nutzt, aber inhärent DSGVO-konform ist. Dieser Ansatz verwendet distanzmessende 3D-LiDAR-Technologie in Kombination mit bildbasierten Deep Learning Verfahren zur Detektion (Lokalisation und Klassifikation) von Personen in 3D-Punktwolken. Im Gegensatz zu Kameras entstehen auf diese Weise keine 2D-Bilder mit visuellen Informationen, sondern 3D-Punktwolken der räumlichen Umgebung, die bei heutigen Auflösungen keine Identifikation von Personen zulassen (s. Abb. 1).

Die folgenden Abschnitte dieses Beitrags sind wie folgt gegliedert. Im nächsten Abschnitt wird ein Überblick über heutige Ansätze zur Objektdetektion in 2D-Bildern und

[1] https://www.datenschutz-grundverordnung.eu/ (abgerufen am 15.09.2022)

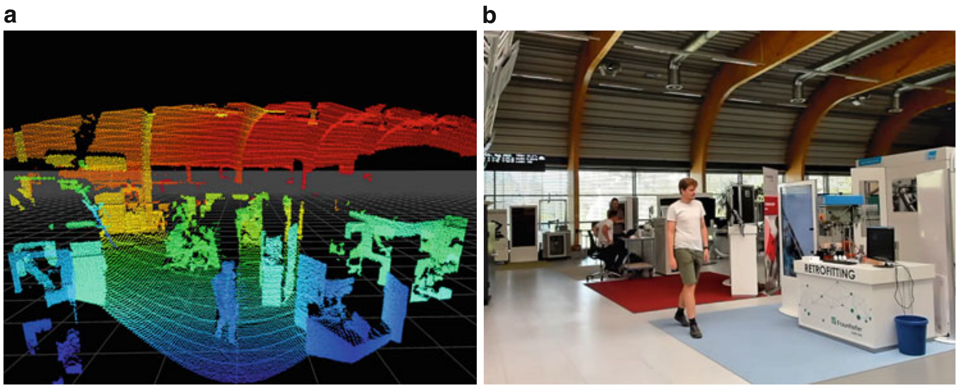


Abb. 1 3D-Punktwolke (**a**) und 2D-Farbbild (**b**) einer Szene

3D-Punktwolken gegeben, bevor das entwickelte DSGVO-konforme Personendetektionsverfahren im darauffolgenden Abschnitt im Detail vorgestellt wird. Dieses wird anschließend im folgenden Abschnitt bzgl. verschiedener Evaluationsmetriken bewertet. Abgeschlossen wird dieser Beitrag mit einem gesamtheitlichen Fazit und einem Ausblick auf weitere Folgeaktivitäten.

2 Stand der Technik

Da bei dem vorgeschlagenen Ansatz 3D-Punktwolken mittels bildbasierten Deep Learning Verfahren verarbeitet werden, wird in den folgenden Unterabschnitten auf den Stand der Technik bei Objektdetektionsverfahren für 2D-Bilder und 3D-Punktwolken eingegangen.

2.1 Objektdetektion in 2D-Bildern

Die Detektion von Objekten in 2D-Bildern hat sich in den letzten 10 Jahren stark weiterentwickelt und hat einen robusten Stand erreicht, der auch in Produkten eingesetzt wird. Der Grund für diesen großen Fortschritt begann mit dem Aufkommen der ersten tiefen neuronalen Netze zur Klassifikation von Bildern zu Beginn des vergangenen Jahrzehnts [4]. Auch wenn die Idee von neuronalen Netzen zur direkten Verarbeitung von Bilddaten (*Convolutional Neural Network*, CNN) zu diesem Zeitpunkt nicht mehr neu war [5], so führten die Verbesserung der verfügbaren (GPU)-Rechenkapazität und die Verfügbarkeit großer annotierter Datensätze [6] zu einem Durchbruch von tiefen neuronalen Netzen. Dies zeigt sich vor allen Dingen durch immer neuere und komplexere Architekturen der neuronalen Netze für die Bildklassifikation, die zu einer stetigen Verbesserung der Erkennungsleistung auf komplexen und herausfordernden Bilddatensätzen führten. Be-

kannte grundlegende Architekturen sind hierbei u. a. VGG [7], GoogleNet/Inception [8], ResNet [9] oder DenseNet [10]. Zudem gab es viele inkrementelle Weiterentwicklungen dieser Architekturen [11, 12], sodass es heutzutage eine sehr gute und robuste Basis für die Bildklassifikation gibt.

Dieser Fortschritt im Bereich der Bildklassifikation mittels Verfahren des Deep Learning führte auch zu einem signifikanten Fortschritt im Bereich der Objektdetektion, bei dem mehrere Objekte in einem Bild sowohl in Form von *Bounding Boxes* lokalisiert als auch klassifiziert werden. Bei diesen Verfahren unterscheidet man generell zwischen zweistufigen und einstufigen Verfahren. Die Familie von R-CNN Architekturen [2, 13, 14] ist die wohl bekannteste Vertreterin von zweistufigen Objektdetektionsverfahren, bei denen zunächst Kandidaten für Objekte auf unterschiedliche Weise generiert und anschließend einzeln klassifiziert werden. Im Gegensatz dazu bestimmen einstufige Verfahren die Bounding Boxes und Klassenzugehörigkeit von Objekten in einem Schritt, ohne dass explizit Kandidatenregionen generiert werden müssen. Hierbei wird ein Backbone-Netz zur Extraktion von *Feature Maps* genutzt (ähnlich wie bei einer Bildklassifikation) und anschließend weitere Schichten zur Bestimmung der Bounding Boxes und Klassenzugehörigkeit angefügt. Bekannte Architekturen dieser Kategorie sind YOLO (und dessen Weiterentwicklungen) [1, 15], SSD [16] und RetinaNet [17]. Heutzutage bilden bildbasierte Objektdetektionsverfahren die Basis für viele verschiedene Anwendungen, u. a. bei der Verkehrsüberwachung, in der Robotik oder in der Industrie. Dies zeigt, dass neuronale Netze zur Objektdetektion auf 2D-Bildern einen hohen Reifegrad erreicht haben.

2.2 Objektdetektion in 3D-Punktwolken

Im Vergleich zur Objektdetektion auf 2D-Bildern ist die Detektion von Objekten in 3D-Punktwolken komplexer und bringt zusätzliche Herausforderungen mit sich. So sind 3D-Punktwolken inhärent unsortiert, nur spärlich besetzt und die Punktdichten unterscheiden sich stark. Diese Effekte entstehen z. B. durch Verdeckungen, Scanmuster oder die effektive Reichweite des Sensors, wobei Punkte in der Entfernung eine geringere Dichte aufweisen als in der Nähe. Ähnlich wie im Bereich der bildbasierten Objektdetektion können hier klassische Ansätze genutzt werden, bei denen Merkmale von Objekten manuell entwickelt werden und für eine anschließende Klassifikation dienen.

Mit dem Aufkommen von Deep Learning Ansätzen im Gebiet der Bildverarbeitung können diese auch für eine 3D-Objektdetektion unter Anpassungen genutzt werden. Derartige Verfahren erfordern strukturierte Daten/Tensoren, z. B. Bilder oder Videos, was jedoch nicht zu den Eigenheiten von Punktwolken gehört, sodass die Verfahren entsprechend adaptiert werden müssen. Qian et al. unterscheiden generell zwischen zwei unterschiedlichen Ansätzen (und einer Kombination aus beiden Ansätzen), um diese Herausforderung zu adressieren [18]: Voxel-basierte Ansätze wandeln die irregulären Punktwolken in reguläre Strukturen um, auf denen dann CNN angewandt werden können. Ein wichtiger Vertreter dieser Kategorie ist das 3D-Detektionsframework VoxelNet, das die Punktwolke

in gleich große Voxel aufteilt, die durch eine einheitliche Merkmalsrepräsentation beschrieben werden und als Basis für eine Objektdetektion dienen [19]. Weitere Vertreter dieser Kategorie sind PointPillars [20], wo eine Punktwolke zunächst in der x-y-Ebene diskretisiert wird und in eine Menge von *Pillars* resultiert, und CenterPoint [21], bei dem auf Basis einer erstellten *top-view* Karte die Objektzentren bestimmt werden. In der zweiten Kategorie von Ansätzen werden Punktwolken direkt verarbeitet, wie es z. B. bei PointNet [22] der Fall ist, das die Basis für weitere Verfahren bildet [23, 24]. Die aktuelle Forschung im Bereich der Objektdetektion in 3D-Punktwolken zeigt, dass diese Verfahren einen enormen Fortschritt machen und hohes Potenzial aufweisen, aber noch nicht den Reifegrad von bildbasierten Objektdetektionsverfahren erreicht haben, insbesondere auch im Hinblick auf Verfügbarkeit von entsprechenden Algorithmen in Open-Source-Software-Bibliotheken oder Unterstützung durch die Community.

Weitere Ansätze kombinieren für bessere Ergebnisse Tiefeninformationen mit Farbbildern [25, 26], jedoch widerspricht dies dem Ziel einer DSGVO-konformen Lösung, die im Fokus dieses Beitrags steht.

3 DSGVO-konforme Personendetektion

Um von der Robustheit bildbasierter Personendetektionsverfahren mittels Deep Learning zu profitieren und gleichzeitig inhärent Aspekte des Datenschutzes zu berücksichtigen, wird in diesem Beitrag ein Ansatz basierend auf distanzmessender 3D-LiDAR-Sensorik in Kombination mit etablierten bildbasierten Deep Learning Verfahren vorgeschlagen. Dies lässt sich als zweistufiges Objektdetektionsverfahren einordnen, bei dem zunächst Objektregionen in der 3D-Punktwolke generiert und anschließend im 2D-Bild klassifiziert werden. Ein Überblick über das entwickelte Personendetektionsverfahren ist in Abb. 2 dargestellt.

3.1 Datenerfassung & Hintergrundentfernung

Zur Datenerfassung wird 3D-LiDAR-Sensorik eingesetzt, die statisch in die Umgebung installiert wird und die Umgebung in Form einer 3D-Punktwolke abbildet. Diese ermöglicht die Detektion von Personen (und anderen Objekten), jedoch keine Identifikation der Personen, wie es z. B. mit hochauflösenden Kameras möglich ist. Daher werden auf diese Weise keine personenbezogenen Daten aufgenommen. Um die zu verarbeitende Datenmenge zu reduzieren, wird das statische Setup des Sensors ausgenutzt, denn die relevanten Objekte (Personen) sind nicht Teil der statischen Umgebung. Hierzu wird der Hintergrund der aktuellen Punktwolke mit Hilfe eines zuvor erstellten Hintergrundmodells abgeglichen und entfernt. Das Hintergrundmodell wird initial über mehrere Frames bei einem statischen Hintergrund erstellt und umfasst somit die statischen Messpunkte der Umgebung, z. B. nicht-bewegliche Maschinen oder statische Strukturen der Fabrikumgebung. Das Er-

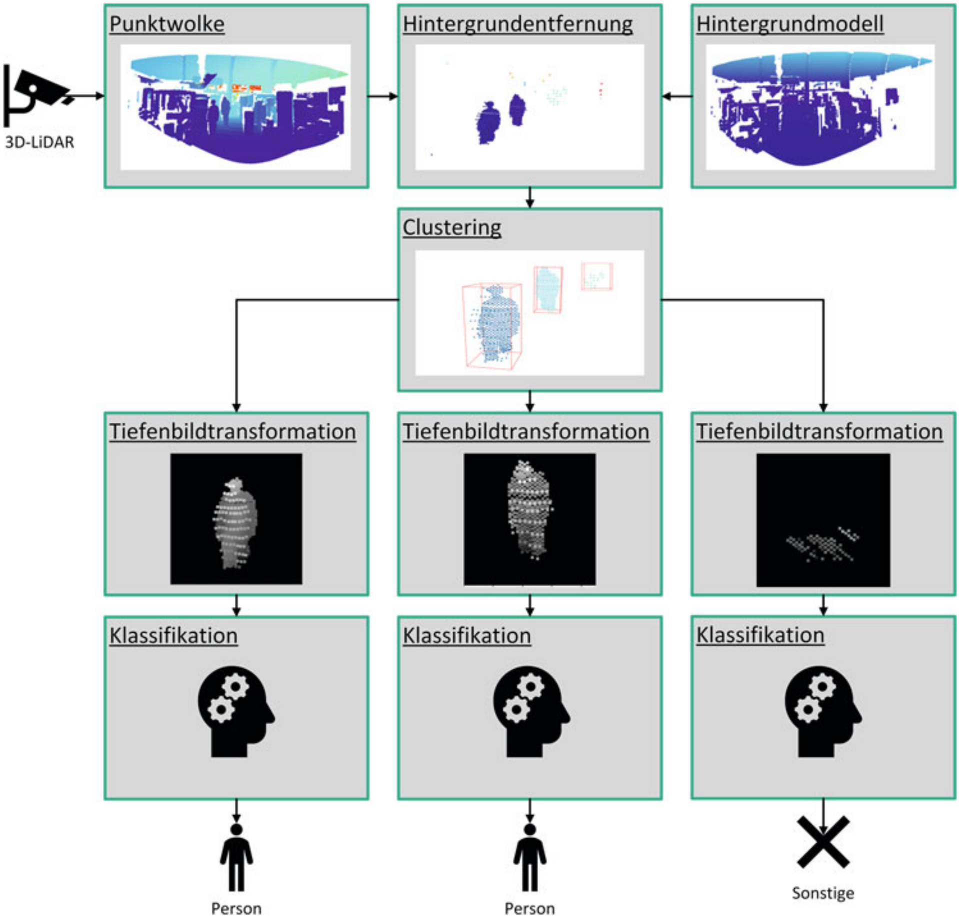


Abb. 2 Visualisierung der Verarbeitungspipeline

gebnis der Hintergrundentfernung ist eine Punktwolke, die nur noch Punkte enthält, die sich nicht im Hintergrundmodell befinden. Dies sind im Fabrikkontext typischerweise sich bewegende Objekte, wie z. B. Personen, Roboterarme oder Fahrerlose Transportfahrzeuge (FTF). Da diese Menge an Punkten im Verhältnis zur gesamten Anzahl an Punkten in der Punktwolke in den meisten Fällen kleiner ist, kann so auch die zu verarbeitende Datenmenge abhängig von der Umgebung signifikant reduziert werden. Dies erleichtert eine spätere Umsetzung des Verfahrens auf einer Embedded Hardware mit limitierten Ressourcen.

3.2 Clustering

Das Ziel des nächsten Verfahrensschritts ist die Generierung von Objektregionen in der 3D-Punktwolke, die anschließend in 2D-Bilder transformiert werden. Dazu werden räumlich naheliegende Punkte mit Hilfe des dichtebasierten Clusteringverfahrens DBSCAN [27] zu Objekten zusammengefasst. Dieser Algorithmus wird genutzt, da er Cluster in beliebiger Form finden kann, die Anzahl der Cluster nicht von vornherein bekannt sein muss und Rauschobjekte erkannt werden können. Das Resultat des Clustering ist eine Menge von Objekten, wobei jedes Objekt wiederum aus einer Menge von Punkten besteht.

3.3 Tiefenbildtransformation & Klassifikation

Aufgrund des Messprinzips eines LiDAR-Sensors lassen sich die Objekte direkt im 3D-Raum lokalisieren, jedoch sind die Objekte noch nicht klassifiziert. Um hier auf etablierte Verfahren des bildbasierten Deep Learning zurückzugreifen, werden die Punktwolken der Objekte mit entsprechenden Eigenschaften im darauffolgenden Schritt in ein 2D-Tiefenbild transformiert, wobei der Grauwert die Distanz zum Sensor angibt. Hierbei wird eine Frontalansicht gewählt, um den charakterisierenden Umriss einer Person bestmöglich zu erfassen. Dabei wird die Punktwolke aus der Sicht des 3D-LiDAR-Sensors betrachtet und der entsprechende Tiefenwert in ein 2D-Bild projiziert. Jedes Tiefenbild, das mit einem Objekt korrespondiert, wird abschließend mittels eines speziell angepassten tiefen neuronalen Netzes klassifiziert. Hierfür wird ein zuvor eigener aufgebauter Datensatz mit Annotationen für das Training des neuronalen Netzes genutzt, das die beiden Klassen *Person* und *Sonstige* berücksichtigt.

4 Evaluation

Zur Evaluation des entwickelten Verfahrens wurde eine mobile Messeinrichtung entworfen, um an verschiedenen Standorten 3D-Punktwolken aufzunehmen. Anschließend wurde ein Bilddatensatz aufgebaut und entsprechend annotiert, sodass dieser für das Training des neuronalen Netzes zur Objektklassifikation genutzt werden konnte. Die Details zu der Evaluation werden in den folgenden Unterabschnitten genauer erläutert.

4.1 Hardware

Die entwickelte Messeinrichtung umfasst u. a. eine Recheneinheit inklusive Datenspeicher, einen WLAN Access Point und einen Akku, um eine mobile und temporäre Datenaufzeichnung zu ermöglichen. Als 3D-LiDAR-Sensor, der an die Messeinrichtung

angeschlossen wird, wird ein Blickfeld Cube 1[2] verwendet. Dieser LiDAR hat eine typische Reichweite von 1,5 m bis 75 m mit einem maximalen Öffnungswinkel vom $72° \times 30°$, sodass ein weites Sichtfeld auf diese Weise abgedeckt werden kann und für einen Fabrikkontext geeignet ist. Die Auflösung der Punktwolke und die Bildwiederholrate lässt sich in Abhängigkeit voneinander konfigurieren. Mit dem Ziel, Personen zu detektieren, wurde der Fokus bei der Parametrisierung des Sensors vornehmlich auf eine hohe Auflösung der Punktwolke und weniger auf eine hohe Bildwiederholrate gelegt. Hierbei wurde eine vertikale Auflösung von 230 Scanlinien und eine horizontale Auflösung von 0,4° bei einer Bildwiederholrate von 2,4 Hz gewählt. Dadurch sollen Details zur Klassifikation eines Objekts als Person erkennbar werden, während die relativ geringe Bildwiederholrate für die Erfassung von Personen bei typischen Geschwindigkeiten von etwa $1{,}5\,\mathrm{m\,s^{-1}}$ ausreichend ist.

4.2 Datenaufnahme & Datensatz

Mittels dieser Messeinrichtung mit angeschlossenem 3D-LiDAR-Sensor wurde ein Datensatz von Punktwolken aufgenommen. Hierzu wurde der Sensor auf einem Stativ in einer Höhe von etwa 4 m mit einer Neigung von 16° an unterschiedlichen Standorten installiert, um einem beispielhaften Aufbau in einer Fabrikumgebung nahezukommen. Diese extrinsischen Kalibrierungsparameter wurden gespeichert und bei der Tiefenbildtransformation genutzt. Anschließend wurden gezielt Punktwolken mit Objekten, insbesondere Personen, aufgezeichnet, wobei auch auf eine möglichst hohe Varianz geachtet wurde. Dies sind Varianten, die auch in (größeren) Fabriken auftauchen können. So wurden u. a. allein gehende Personen, Personengruppen, Personen mit Koffern, Fahrrädern oder Schiebewagen mit aufgezeichnet. Die Personen hatten bei der Datenaufnahme eine Distanz von bis zu 25 m zum Sensor. Ein Betreuer des Messaufbaus war während der Datenakquise permanent anwesend und hat sich zu den aufgezeichneten Punktwolken die entsprechenden Objektklassen notiert.

Jede der aufgezeichneten Punktwolken wurde anschließend mit Hilfe des in Kap. 3 beschriebenen Verfahrens bis zur Erstellung eines Tiefenbildes pro Objekt verarbeitet. Auf diese Weise ist ein Bilddatensatz mit etwa 30K Bildern mit einer Auflösung von 224×224 Pixel entstanden[3]. Jedes dieser Bilder wurde manuell mit einer der beiden zu berücksichtigen Klassen annotiert: *Person* und *Sonstige*. Ein Überblick über Beispielbilder der beiden Klassen ist in Abb. 3 visualisiert. Während die Klasse *Person* alle Bilder mit Personen enthält, umfasst die Klasse *Sonstige* alle vom Hintergrund extrahierten Objekte, die keine Person darstellen, z. B. Teile von sich bewegenden Objekten.

[2] https://www.blickfeld.com/lidar-sensor-products/cube-1/ (abgerufen am 15.09.2022)
[3] Eine Bildauflösung von 224×224 Pixel ist typisch für CNN-basierte Klassifikationsnetze und historisch bedingt [4, 7, 8].

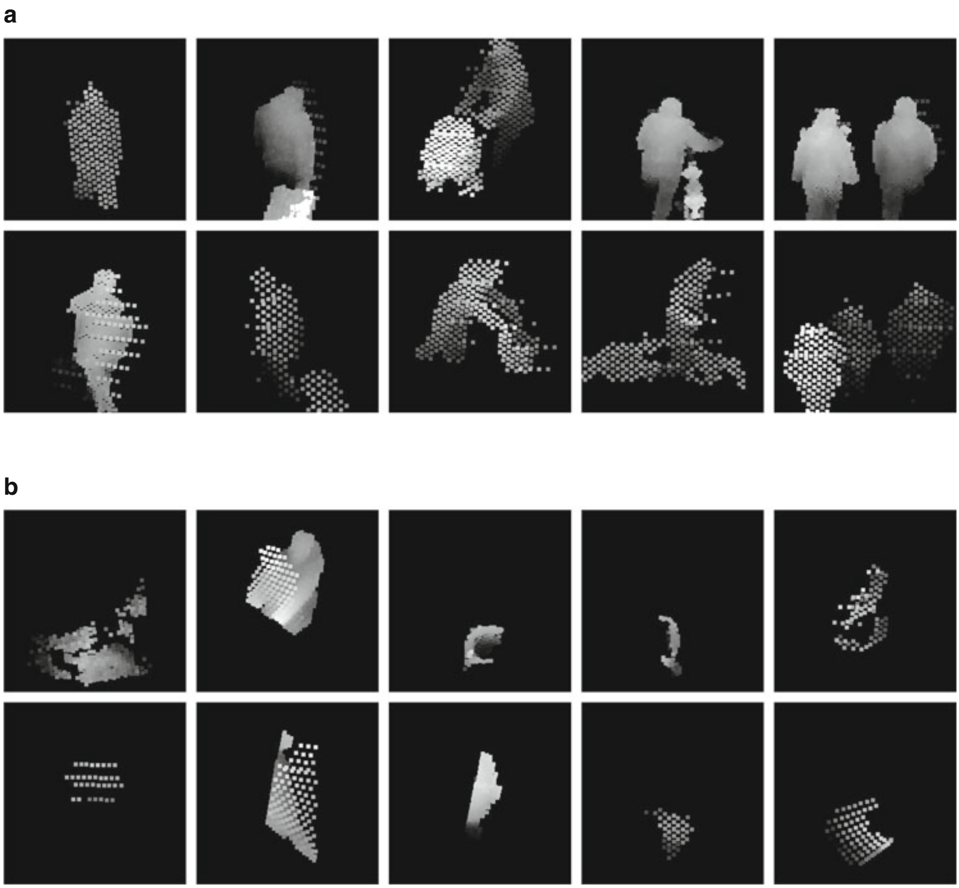


Abb. 3 Beispielbilder aus dem erstellten Bilddatensatz mit den beiden Klassen *Person* (**a**) und *Sonstige* (**b**) für das Training des neuronalen Netzes

4.3 Training

Dieser annotierte Bilddatensatz wurde anschließend genutzt, um ein tiefes neuronales Klassifikationsnetz zu trainieren. Hierzu wurde der Datensatz zunächst in einem Verhältnis von 70:10:20 in einen Trainings-, Validierungs- und Testdatensatz aufgeteilt. Um das neuronale Netz robuster gegenüber bestimmten Transformationen zu machen, wurden die Bilddaten zufällig augmentiert (Flip, Translation, Rotation). Als Architektur des neuronalen Netzes wurde ein ResNet-50 [9] verwendet, das eine state-of-the-art Architektur für neuronale Netze zur Bildklassifikation ist. Dies besitzt zwar weniger Parameter als andere Architekturen, z. B. ResNet-101 [9] aus derselben Familie, ist jedoch für das vorliegende binäre Problem ausreichend komplex und hat zudem positive Effekte auf Speicherbedarf und Inferenzzeit. Als Loss-Funktion wurde die Kreuzentropie zwischen den tatsächlichen

Klassen des Bilddatensatzes und der Ausgabe des neuronalen Netzes berechnet. Zur Optimierung der Parameter des neuronalen Netzes während des Trainingsprozesses wurde der Adam-Optimierer [28] mit einer Lernrate von 0.001 genutzt. Die Eingabebilder hatten eine Auflösung von 224 × 224 Pixel und wurden in Batchgrößen von 64 Bildern bereitgestellt. Diese entstammten dem Trainingsdatensatz mit etwa 21K Bildern für die Optimierung der Zielfunktion und dem Validierungsdatensatz mit etwa 3K Bildern zur Bestimmung des Loss am Ende jeder Epoche. Der Trainingsprozess fand über 100 Epochen auf zwei NVIDIA Tesla V100 Tensor-Recheneinheiten statt.

4.4 Ergebnisse

Nach dem Training des neuronalen Netzes wurde dessen Güte auf dem Testdatensatz mit etwa 6K Bildern (20 % des Bilddatensatzes) evaluiert. Diese Bilddaten waren nicht Teil des Trainingsprozesses, sodass diese für das neuronale Netz neu waren. Die quantitativen Ergebnisse dieser Auswertung sind in Tab. 1 zusammengefasst. Insgesamt erreicht das neuronale Netz für dieses binäre Klassifikationsproblem eine Accuracy von 98 %, was auf eine sehr gute Güte hinweist. Fehlklassifikationen können entstehen, wenn sich die Bilder beider Klassen stark ähneln, z. B. bei weit entfernten Objekten, die aufgrund der geringeren Punktdichte in der Entfernung keine entsprechenden Merkmale mehr aufweisen.

Das gute Ergebnis zeigt sich auch in der Visualisierung der Personendetektion in einer 3D-Punktwolke, die in Abb. 4 dargestellt ist. In dieser sind zwei Personen zu sehen, die

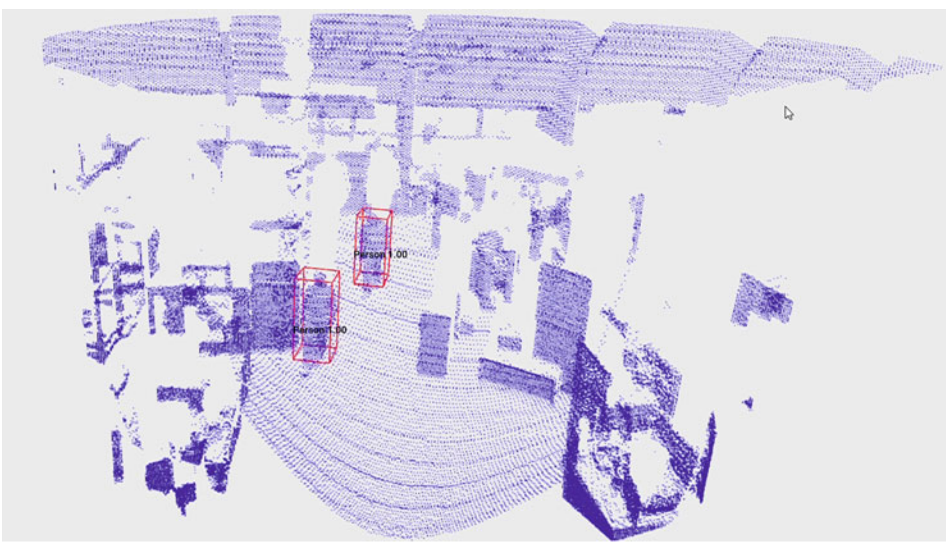


Abb. 4 Visualisierung der Personendetektion in einer 3D-Punktwolke

Tab. 1 Evaluationsmetriken des neuronalen Netzes auf dem Testdatensatz

	Precision	Recall	F1-Score	Anzahl
Sonstige	0,97	0,99	0,98	2594
Person	1,00	0,98	0,99	3469
Gewichtetes Mittel	0,98	0,98	0,98	6063
Accuracy			0,98	6063

detektiert und korrekterweise als Personen klassifiziert werden. Die restlichen Punkte der Umgebung werden richtigerweise nicht als Objekte bestimmt.

5 Fazit und Ausblick

Die Ergebnisse dieses Beitrags zeigen, dass es mit Hilfe des entwickelten Verfahrens möglich ist, Personen robust in 3D-Punktwolken zu detektieren und von anderen Objekten zu unterscheiden. Im Gegensatz zu etablierten Objektdetektionsverfahren basierend auf (hochaufgelösten) Farbbildern werden bei dem vorgestellten Ansatz aufgrund seines Messprinzips keine personenbezogenen Daten verarbeitet, sodass dieser Ansatz an sich alleine in Bezug auf den Datenschutz unkritisch zu sehen ist. Zudem liefert dieser Ansatz im Vergleich zu kamerabasierten Ansätzen inhärent Tiefeninformationen der Szene, die für eine 3D-Positionsbestimmung der Objekte direkt genutzt werden können. Weiterhin müssen Personen keine Gegenstände, wie z. B. drahtlose IoT-Devices, bei sich tragen, um in der Umgebung lokalisiert zu werden. Solch ein Ansatz ist gut dafür geeignet, im Rahmen einer Fabrikumgebung eingesetzt zu werden, um u. a. die Sicherheit von Arbeitern zu erhöhen oder das Fabriklayout zu optimieren.

Zukünftig soll das entwickelte Verfahren auf einer Embedded Hardware umgesetzt und als prototypisches System in der SmartFactoryOWL[4] evaluiert werden. Hierbei bietet es sich zur Optimierung der Verarbeitung an, vorverarbeitende Schritte des Verfahrens, wie z. B. die Hintergrundentfernung, direkt in den 3D-LiDAR-Sensor auszulagern. Zusätzlich sollen die Detektionen im 3D-Raum zeitlich verfolgt werden (*Tracking*), um Bewegungsmuster von Personen zu bestimmen und höherwertige Informationen abzuleiten. Zudem ist dieser Ansatz nicht nur auf einen Fabrikkontext beschränkt, sondern kann auch in anderen Domänen genutzt werden, um DSGVO-konform Personen zu detektieren. Beispielsweise wurde im Rahmen des Projekts „KI4PED“ der vorgeschlagene Ansatz zur Erfassung von Personen im Straßenverkehr mit dem Ziel einer Optimierung der Fußgängerüberquerungszeiten an Lichtsignalanlagen erprobt.

Danksagung Dieser Beitrag entstand im Rahmen des Projekts „KI4PED“ (FKZ: 19F1090A), das im Rahmen der Innovationsinitiative mFUND durch das Bundesministerium für Digitales und Verkehr (BMDV) gefördert wurde.

[4] https://smartfactory-owl.de/ (abgerufen am 15.09.2022)

Literatur

1. Bochkovskiy A, Wang CY, Liao HYM (2020) YOLOv4: Optimal speed and accuracy of object detection (arXiv preprint arXiv:2004.10934)
2. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 28:91–99
3. Silva B, Pang Z, Akerberg J, Neander J, Hancke G (2014) Experimental study of UWB-based high precision localization for industrial applications. In: IEEE International Conference on Ultra-WideBand (ICUWB), S 280–285
4. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems (NIPS), S 1097–1105
5. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
6. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 248–255
7. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR), S 1–14
8. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 1–9
9. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 770–778
10. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 2261–2269
11. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI Conference on Artificial Intelligence. AAAI, Palo Alto, S 4278–4284
12. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 5987–5995
13. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 580–587
14. Girshick R (2015) Fast R-CNN. In: IEEE International Conference on Computer Vision (ICCV), S 1440–1448
15. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 779–788
16. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: single shot multibox detector. In: European Conference on Computer Vision (ECCV), S 21–37
17. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: IEEE International Conference on Computer Vision (ICCV), S 2999–3007
18. Qian R, Lai X, Li X (2022) 3D object detection for autonomous driving: a survey. Pattern Recognit 130:1–19

19. Zhou Y, Tuzel O (2018) VoxelNet: end-to-end learning for point cloud based 3D object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), S 4490–4499
20. Lang AH, Vora S, Caesar H, Zhou L, Yang J, Beijbom O (2019) PointPillars: fast encoders for object detection from point clouds. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), S 12689–12697
21. Yin T, Zhou X, Krähenbühl P (2021) Center-based 3D object detection and tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), S 11779–11788
22. Charles RQ, Su H, Kaichun M, Guibas LJ (2017) PointNet: deep learning on point sets for 3D classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 77–85
23. Qi CR, Yi L, Su H, Guibas LJ (2017) PointNet++: deep hierarchical feature learning on point sets in a metric space (arXiv preprint arXiv:1706.02413)
24. Shi S, Wang X, Li H (2019) PointRCNN: 3D object proposal generation and detection from point cloud. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), S 770–779
25. Gonzalez A, Villalonga G, Xu J, Vázquez D, Amores J, Lopez AM (2015) Multiview random forest of local experts combining RGB and LIDAR data for pedestrian detection. In: Intelligent Vehicles Symposium (IV), S 356–361
26. Simon M, Amende K, Kraus A, Honer J, Sämann T, Kaulbersch H, Milz S, Gross HM (2019) Complexer-YOLO: real-time 3D object detection and tracking on semantic point clouds. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), S 1190–1199
27. Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: International Conference on Knowledge Discovery and Data Mining (KDD). AAAI, Palo Alto, S 226–231
28. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR), S 1–15

Advanced Feature Extraction Workflow for Few Shot Object Recognition

Markus Brüning, Paul Wunderlich, and Helene Dörksen

Abstract

Object recognition is well known to have a high importance in various fields. Example applications are anomaly detection and object sorting. Common methods for object recognition in images divide into neural and non-neural approaches: Neural-based concepts, e.g. using deep learning techniques, require a lot of training data and involve a resource intensive learning process. Additionally, when working with a small number of images, the development effort increases. Common non-neural feature detection approaches, such as SIFT, SURF or AKAZE, do not require these steps for preparation. They are computationally less expensive and often more efficient than the neural-based concepts. On the downside, these algorithms usually require grey-scale images as an input. Thus, information about the color of the reference image cannot be considered as a determinant for recognition. Our objective is to achieve an object recognition approach by eliminating the "color blindness" of key point extraction methods by using a combination of SIFT, color histograms and contour detection algorithms. This approach is evaluated in context of object recognition on a conveyor belt. In this scenario, objects can only be recorded while passing the camera's field of vision. The approach is divided into three stages: In the first step, Otsu's method is applied among other computer vision algorithms to perform automatic edge detection for object localization. Within the subsequent second stage, SIFT extracts key points out of the previously identified region of interest. In the last step, color histograms of the specified region

M. Brüning (✉) · P. Wunderlich · H. Dörksen
inIT – Institute Industrial IT, Technische Hochschule Ostwestfalen-Lippe, Lemgo, Deutschland
e-mail: markus.bruening@th-owl.de

P. Wunderlich
e-mail: paul.wunderlich@th-owl.de

H. Dörksen
e-mail: helene.doerksen@th-owl.de

V. Lohweg (Hrsg.), *Bildverarbeitung in der Automation*, Technologien für die intelligente Automation 17, https://doi.org/10.1007/978-3-662-66769-9_4

are created to distinguish between objects that feature a high similarity in the extracted key points. Only one image is sufficient to serve as a template. We are able to show that developing and applying a concept with a combination of SIFT, histograms and edge detection algorithms successfully compensates the color blindness of the SIFT algorithm. Promising results in the conducted proof of concept are achieved without the need for implementing complex and time consuming methods.

Keywords

Object recognition · SIFT · Color histograms · Computer vision · Few shot

1 Introduction

Recognizing objects based only on few or even one samples gained a lot of attention in recent years and is currently a hot topic in computer vision and machine learning [1]. There are numerous fields of application in which recognizing objects based on few images is desired and already under investigation: In dermatological disease diagnosis within the medical domain few-shot learning is applied to support doctors based on few given examples [2]. In the agriculture domain, the classification of healthy and diseased plants is of crucial importance as it preserves and improves the yield [3].

Achieving fast and reliable object recognition having only one or few images is also of interest for industrial applications: In case of customized products in small batch series production with reaching a “lot-size-of-one” only very few images can be taken after assembly [4]. In such a case, images of the customized items cannot be provided in the run-up to learn a deep-learning based classifier.

There are several reasons why in some cases only few data exists [1]:

1. Imitate the way humans learn: Only providing that much data that a human would require
2. Cases in which events only rarely occur
3. Reducing the amount of data subsequently reduces the data gathering effort as well as the computational cost

The target use-case of this work is object recognition applied in a conveyor belt system. A concept drawing of the aforementioned conveyor belt example is shown in Fig. 1: A set of objects A, B, C, … moves along a conveyor with an off-the-shelf camera mounted on top of it. While the object travels on the belt it passes the cameras field of vision. During this time, images of the object can be recorded. In a second run, detected objects are checked for similarity with the previously recorded set of objects: It can be detected which objects have been recorded before and their position can be estimated.

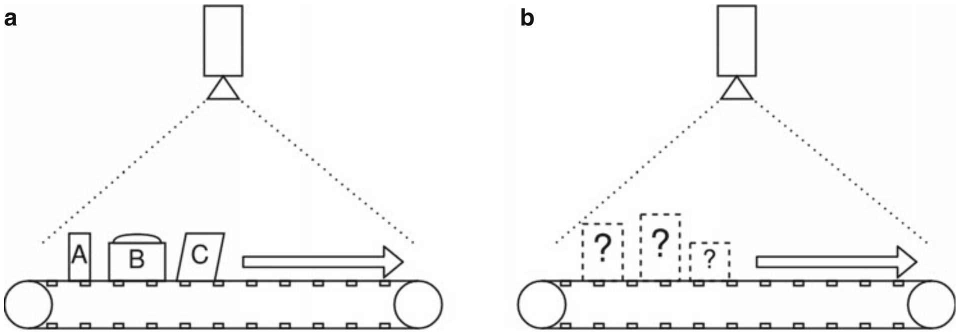


Fig. 1 The baseline scenario for this work: A "learning-phase" (**a**) and a "recognition-phase" (**b**).

1.1 State of the Art

The resulting major challenge of this scenario is the small number of images that can be recorded. Object detection methods in general can be divided into two architectural approach categories:

- **Neural:** Deep-learning methods, one- and two-stage detectors like RCNN [5], YOLO [6] and SSD [7] [8].
- **Non-neural:** Histogram of Oriented Gradients (HOG) [9], the Viola Jones Detector [10], Scale Invariant Feature Transform (SIFT) [11] and others [8].

Neural and non-neural approaches both have individual advantages and disadvantages: Non-neural methods do not have a requirement for training data or complex neural networks. On the other hand, the processing time for decision making might be longer [12]. Neural approaches however might deliver more accurate results (especially on challenging backgrounds) but require a larger training dataset [12].

When working with few or even only one image(s) common off-the-shelf deep learning methods cannot be applied due to the fact that deep learning does not perform well on smaller datasets [13]. The challenge of building a classifier only based on very few images is called *few-shot learning*. A related learning problem for this task category is called transfer learning in which knowledge is transferred from a domain with has sufficient data available [1]. In addition it can be checked if the dataset can be artificially enlarged using *data augmentation* which adds different kinds of invariance to the available images.

Regarding the non-neural based approaches, popular feature extraction systems like SIFT [11], SURF [14] or AKAZE [15] have a common drawback: Besides the ability of successfully identifying and localizing distinctive features in images, many methods of this category require grey-scale images as an input for further processing. This obviously abstracts away valuable information about the coloration present in an image.

Especially the neural based methods gained a lot of interest and development in recent years. Specialized few-shot learning (FSL) [1] and one-shot learning (OSL) [13] approaches made great progress but increased the complexity and engineering effort. The first idea of one-shot learning has been investigated by [16] in 2006. There are several approaches to tackle few-shot learning applications. They usually require some kind of prior knowledge which is used on different perspectives like the *data*, the *model* or the *algorithm* [1].

1.2 Related Work

The color-blindness of feature description algorithms like SIFT is no novelty in research and has been similarly investigated before: Suhasini et al. presents an approach for image retrieval using Invariant Color Histograms. The authors use the HSV instead of the RGB color space [17]. In [18] Chang et al. uses color coocurrence histograms (CH) for recognizing objects in images. Color-CH give information about the separation distance of pairs of colored pixels. This is an addition to normal color histograms, as these do not contain information about geometry features. The authors show successful object recognition on cluttered background, partial occlusions and flexing of the object. Ancuti and Bekaert identified that SIFT has proven to be the most reliable descriptor but is vulnerable to color images [19]. In this work color coocurrence histograms are also used combined with the SIFT approach. The results in context of image matching outperform the original version, detecting an additional number of correct matched feature points.

1.3 Research Question

The research question and the subsequent aim of this work is how a simple object recognition system can be realized without using prior knowledge. Regarding the usage of SIFT this paper evaluates a method for extending SIFT with using coloration information as an additional deciding factor.

To pick up the conveyor belt example from above (shown in Fig. 1) the following challenges are identified:

1. *Few images:* Due to the short recording time on the conveyor belt
2. *Plain background*
3. *Low variation:* Objects are only visible from one viewpoint
4. *Unknown class of objects:* No dataset or prior knowledge from related problems is available

The system should efficiently recognize objects in plain images by only providing few or one image as a template. Due to the usage of established image processing the system

is able to run on hardware with low computational power, instead of requiring expensive hardware components like GPUs.

In order to investigate the questions and requirements, the following chapter proposes an approach by presenting the concept and details of an implementation. The subsequent chapter contains an experiment on a test-dataset and states its results. The last chapter concludes the work and gives an outlook on the topic.

2 Approach

2.1 Concept

To estimate the distinctive textural and shape features of an object present in an image, we choose the well established *SIFT algorithm*. It is an object recognition system that uses local images features which are invariant to scaling, translation, rotation and partially invariant to illumination changes [11]. Reasons for choosing this algorithm are superior results in comparative analysis [20].

An additional tool is used for the object recognition. The creation of *color histograms* represents the pixel-wise color distribution within an image.

The main steps in the presented workflow are depicted in Fig. 2. It represents a shortened version of the full workflow of Fig. 5:

In the **image capture and preprocessing** step the input image is taken by a commercially available camera with a resolution of 640x480 px. The choice of the camera type is arbitrary, as long as the image of the saved reference objects have been taken with the same camera to match the resolution and possible coloration shifts. A standard USB-webcam, a smartphone camera as well as a virtual image feed have been tested as input devices. The preprocessing separates the object's fore- and background of the image and creates a binary mask. This region-of-interest (ROI, the area containing the object) masks out the part of the image that is irrelevant for detection. Then, key point-descriptors of the ROI are **extracted using SIFT**. The found descriptors are subsequently matched with the available templates. This is the first deciding factor for classification. If there are multiple objects that feature a high similarity (from now on called "candidates"), the decision is ambiguous and a **color histogram** of the ROI is created. It is similarly compared with the template images. Thus, the histograms serve as an "arbiter". The final **decision** or assignment is firstly based on the result of SIFT and in a case of multiple candidates the result of the histogram comparison is made use of.

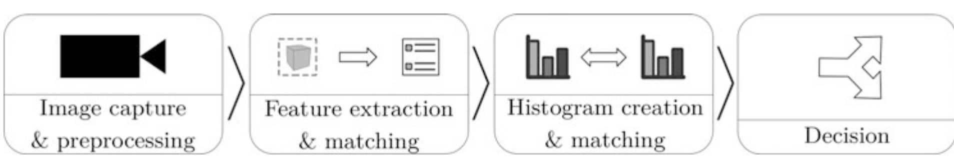


Fig. 2 Summarized programming flowchart.

2.2 Detailed description

The following description refers to Fig. 5. Text in bold notation points to the headings on the right-hand side. The workflow starts with the **preprocessing and image capture** including an initialization of the "known" objects. These are images of objects that have been captured before and are stored as image files. For every object a binary mask is created as explained before. A color histogram of the area provided by the mask is created. This is done for later use before the recognition loop to reduce processing time while detecting. As the color of the background is likely to be captured by the histogram, the corresponding color components have to be excluded from every objects histogram. This is achieved by creating a mask containing only the area of the object. This eliminates the capturing of unnecessary pixels.

The effect of not masking the color components of the background tested is demonstrated on three images shown in Fig. 3. A reference object (a) is compared with a rotated and translated representation of the object (b). Image (c) shows a similar object with a slight color-variation in some parts. The results of Table 1 show the histogram similarity derived from the calculated distance.

After the successful masking the loop is entered starting with image capturing. This begins with receiving an image from a simple USB-camera for example. The contour detection now tries to detect objects within the image. A successful detection provides the region-of-interest which contains the object. If none is found, the loop is iterated-through until a ROI is found. To precisely locate the object, the boundaries and contour of the object have to be located. Therefore, the contour is extracted by applying a method of topological structural analysis using border following [21]. From the hierarchical output

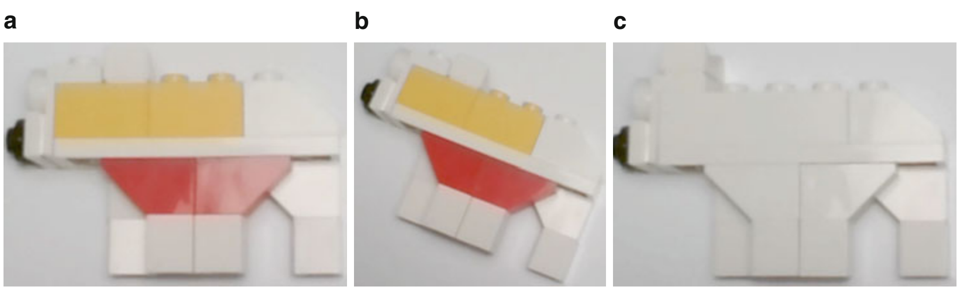


Fig. 3 (**a**) Reference image, (**b**) Reference image rotated and translated, (**c**) Differently colored object.

Table 1 Comparison and Similarity without masking

(a) compared to ...	(b)	(c)
Without masking	48%	47%
With masking	56%	30%

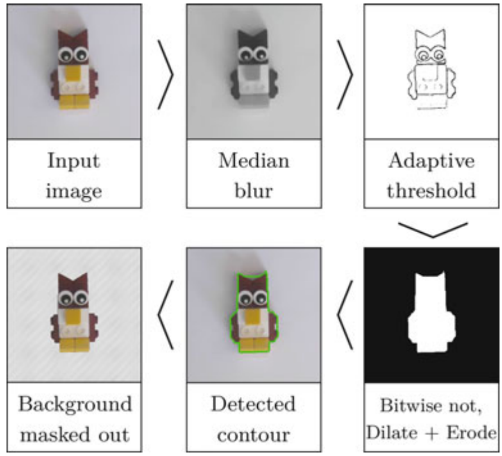


Fig. 4 The image preparation workflow.

only the outermost contour (the “parent”) is used to limit the area the object appears in. The technical details of this image preparation is depicted in Fig. 4.

In detail, the image preparation workflow of Fig. 4 is realized as follows: The input image, in this case the owl figure, is converted to greyscale and a blurring filter is applied for noise reduction. Then, adaptive thresholding is used to extract the edges of the object. Methods without an adaptive property, like Canny Edge Detection [22], are prone to require a manual setting of parameters in order to detect the edges properly. The output of the thresholding step is subsequently inverted via a bitwise-not function. The application of two morphological operations, namely dilating and eroding, again reduces noise. The resulting image distinguishes the objects’ area (indicated by white pixels) from the background (represented by black pixels). Up to this point, this black-and-white image represents a mask dividing fore- and background. The topological structural analysis using border following [21] is now easily applied on the prepared image. The outermost contour (the “parent”) gives information about the objects border that is used to create the bounding-box. Therefore, the smallest and greatest x- and y-coordinates of the detected contour form the top-left and the bottom-right corner of the box.

The **feature extraction and matching** is performed using SIFT. The extraction procedure is restricted to the region-of-interest provided by the bounding box of the masked area. If only few (due to noise) or no features were found by SIFT it is assumed that no object is present in the region. Otherwise, the feature descriptors are matched with the ones from the list of known objects. The matches are stored in a “score list”. This list is subsequently sorted ascending with the highest scores.

Now, candidates are appointed with the requirement of featuring a similarity of at least 50% in order to make a **decision**. This parameter is defined as similar and is chosen freely. If no candidate has been nominated, the object is seen as “unknown”. If there is only one candidate it is a distinct decision. The case of having multiple objects (≥ 2) sharing a high similarity estimated by SIFT is determined by analyzing the coloration-in-

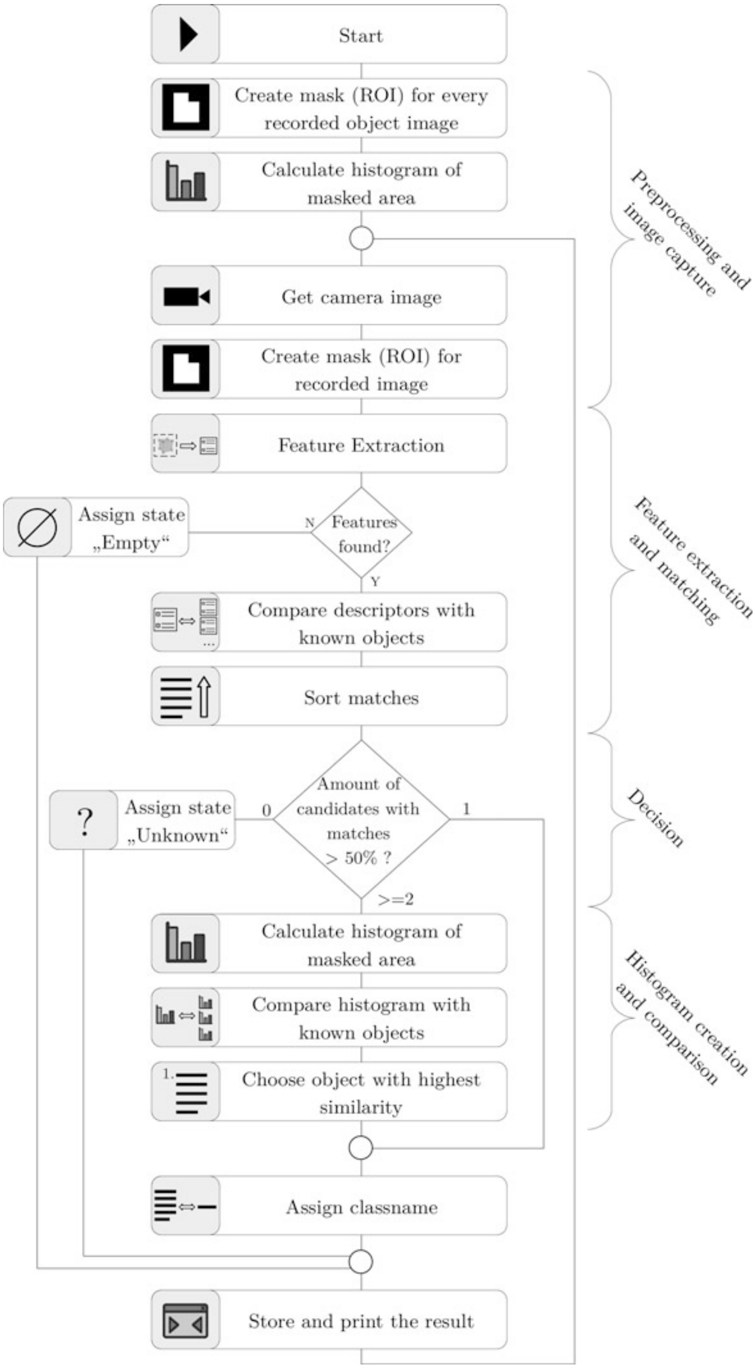


Fig. 5 Extended programming flowchart.

formation. Therefore, a **color histogram** of the recent image is created with the additional background mask as seen before. The histogram of the current image is compared with the ones calculated in the beginning and the results are stored in a list. This list is also sorted based on the scoring. Now, the object corresponding to highest score is estimated to be the match for the newly seen object.

2.3 Concept Drawing

See Fig. 5.

3 Experiments and Results

To validate the added color-variance of the SIFT algorithm and the overall functionality within an object recognition system a proof-of-concept is conducted. The objects themselves used in this context are small Lego® figures. They are originally "produced" in the SmartFactoryOWL[1] to demonstrate the workflow of a cyber-physical-system. For this scenario it is assumed that the bricks of the figures can be chosen in individual ways to fit a customers need.

Fig. 6 shows an overview of the dataset used in the experiment.

The dataset consists of a sum of 30 images representing 10 classes. Each class is captured three times: One image as depicted in Fig. 6 and two images slightly shifted and rotated by 45° and 180°. Four classes within the dataset are additionally present with minor color changes. This is done to challenge the recognition system: These objects a likely

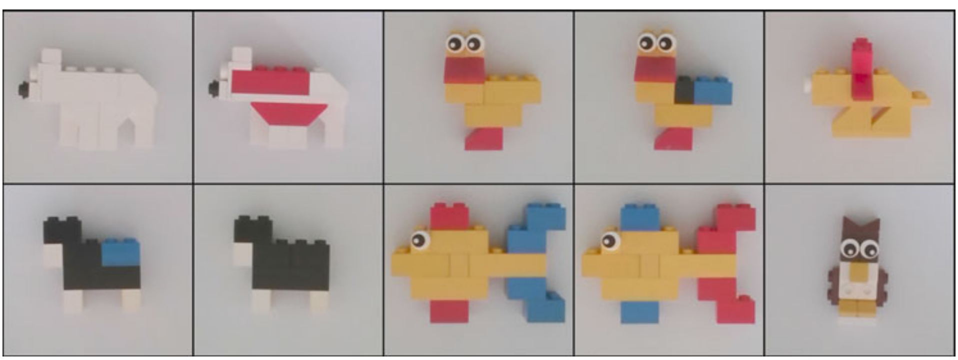


Fig. 6 An image of every class of the dataset (from left to right, top to bottom): A polar bear (2), duck (2), lion, sheep (2), fish (2) and an owl.

[1] https://smartfactory-owl.de

	bear_a	bear_b	duck_a	duck_b	fish_a	fish_b	lion	owl	sheep_a	sheep_b
bear_a	**46**	**40**	15	12	17	21	18	16	22	18
bear_b	**25**	**41**	18	12	16	17	13	15	22	20
duck_a	14	12	**32**	**38**	25	28	12	17	20	13
duck_b	17	12	**32**	**36**	22	27	11	13	19	17
fish_a	19	15	17	16	**35**	**43**	21	22	21	22
fish_b	20	15	17	16	**39**	**32**	17	22	22	20
lion	20	14	18	17	21	18	36	25	23	27
owl	12	6	11	11	11	14	11	45	8	9
sheep_a	28	21	21	23	29	35	22	27	**37**	**45**
sheep_b	26	22	22	22	26	30	23	29	**48**	**41**

Fig. 7 A similarity matrix of SIFT applied on the dataset. The objects refer to Fig. 6. Suffix "_a" denotes a normally colored object and "_b" a variant with slightly altered colors. Results of these objects are framed in a black box. All similarities in percent.

to look similar when observed in grey-scale, but show variations when analyzing the coloration.

The results are represented in form of a classification results matrix. Every object image is compared to every other object. The matrix entries represent similarities and are calculated as follows:

$$\text{SIFTsimilarity} = (\text{KeypointMatches}/\text{TotalKeypoints}) \quad (1)$$

$$\text{HistogramSimilarity} = (1 - \text{HistogramDistance}) \quad (2)$$

$$\text{Similarity} = (\text{SIFTsimilarity} + \text{HistogramSimilarity})/2 \quad (3)$$

The first matrix depicted in Fig. 7 shows how SIFT performs on the provided dataset. The calculated similarities between the objects with color variations (_a and _b) are generally very close but the highest similarity often points to the wrong object leading to a false classification. The difference towards the other classes is sufficient in order to tell these apart.

Evaluating the results of applying a combination of SIFT and color histograms reveals more distinctive decisions in the matrix of Fig. 8. The classes with the color variant feature a higher distance towards each other. The matrix shows that in every case the highest similarity belongs to the correct class, even though rather closely for some cases. The boundary towards the different classes is more distinctive as well. This is indicated by the more reddish coloration within the matrix.

	bear_a	bear_b	duck_a	duck_b	fish_a	fish_b	lion	owl	sheep_a	sheep_b
bear_a	**71**	**47**	19	17	17	19	19	32	30	27
bear_b	**40**	**69**	30	27	25	22	23	30	27	25
duck_a	18	27	**63**	**53**	49	44	48	33	21	16
duck_b	20	27	**50**	**65**	50	52	34	34	26	28
fish_a	18	25	45	47	**66**	**62**	46	31	18	22
fish_b	18	22	38	46	**60**	**65**	38	32	21	25
lion	20	24	51	38	46	39	65	33	18	19
owl	30	26	30	33	26	28	26	70	36	32
sheep_a	33	27	21	28	22	28	17	46	**75**	**59**
sheep_b	31	26	21	31	24	30	16	42	**60**	**72**

Fig. 8 A similarity matrix of the presented workflow applied on the dataset. All similarities in percent.

4 Conclusion and Outlook

Although the used algorithms are rather old in terms of image processing approaches, they have proven to still be useful and beneficial for the evaluated are of application.

In general, working with a low amount of images, in the "few-shot learning" domain, is still a relatively new topic in machine learning. Common state-of-the-art methods do not perform well on smaller datasets and especially may have problems with uni-colored backgrounds due to the risk of overfitting.

Additionally, many approaches in few-shot learning require some kind of prior knowledge, for example regarding the data, model or algorithm [1]. Therefore, a detailed analysis of the environment by an expert is required in order to investigate the availability of similar datasets. All in all, using neural-approaches often results in lots of engineering to find the right models and parameters. We may see further development in the future.

The experiment conducted in this work shows a significant improvement compared to a SIFT-only-based classification. The "challenges"[2] included in the dataset resulted in a low number or ambiguous matches when only SIFT is applied. The proposed method of this work increased the number of correctly classified objects compared in two similarity matrices. The effect on the reduced dataset approaches zero, as it only includes heterogeneous objects which SIFT can successfully distinguish.

But the results also state that using histograms in addition to key point detection is no cure-all solution to determining small variations in color. Little variations in the color components due to illumination changes or other influences during recording can decrease the number of correctly classified objects. This is likely to occur in this case, as no professional equipment was used for recording.

[2] Objects featuring a high degree of similarity but slightly differently colored

On the upside, the proposed classification workflow was achieved by using only lightweight methods without the need of a training stage or a dataset for learning purposes. This is especially attractive for the usage on resource limited hardware. All in all, the workflow presented in this work offers several advantages towards deep-learning methods but offers room for improvement in detecting small coloration changes.

References

1. Wang Y, Yao Q, Kwok JT, Ni LM (2021) Generalizing from a few examples. ACM Comput Surv 53:1–34
2. Prabhu V, Kannan A, Ravuri M et al (2019) Few-shot learning for dermatological disease diagnosis. In: Proceedings of the 4th Machine Learning for Healthcare Conference 106, S 532–552
3. Nuthalapati SV, Tunga A (2021) Multi-domain few-shot learning and dataset for agricultural applications. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)
4. Jasperneite J, Hinrichsen S (2015) Wandlungsfähige Montagesysteme für die Fabrik der Zukunft. In: VDI-Tagung "Industrie 4.0" (Vortrag)
5. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition
6. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
7. Liu W, Anguelov D, Erhan D et al (2016) Ssd: Single shot multibox detector. In: European conference on computer vision
8. Zou Z, Shi Z, Guo Y, Ye J (2019) Object detection in 20 years: A survey. http://arxiv.org/abs/1905.05055
9. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). https://doi.org/10.1109/cvpr.2005.177
10. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001
11. Lowe DG (1999) Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision
12. Subhashini D, Dutt SI (2020) A review on road extraction based on neural and non-neural networks. Int J Eng Res V9:1306–1309
13. Jadon S, Garg A (2020) Hands-on one-shot learning with python: Learn to implement fast and accurate deep learning models with fewer training samples using pytorch. Packt, Birmingham
14. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (surf). Comput Vis Image Underst 110:346–359
15. Alcantarilla P, Nuevo J, Bartoli A (2013) Fast explicit diffusion for accelerated features in nonlinear scale spaces. In: Procedings of the British Machine Vision Conference 2013
16. Fei-Fei L, Fergus R, Perona P (2006) One-shot learning of object categories. IEEE Trans Pattern Anal Machine Intell 28:594–611
17. Suhasini PS, Krishna KS, Krishna IV (2012) Combining sift and invariant color histogram in HSV space for deformation and viewpoint invariant image retrieval. In: 2012 IEEE International Conference on Computational Intelligence and Computing Research

18. Chang P, Krumm J (1999) Object recognition with color cooccurrence histograms. In: Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat No PR00149)
19. Ancuti C, Bekaert P (2007) SIFT-CCH: increasing the SIFT distinctness by color co-occurrence histograms. In: 2007 5th International Symposium on Image and Signal Processing and Analysis
20. Tareen SA, Saleem Z (2018) A comparative analysis of SIFT, surf, Kaze, AKAZE, Orb, and brisk. In: 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)
21. Suzuki S, Abe K (1985) Topological structural analysis of digitized binary images by border following. Comput Vis Graph Image Process 29(3):396
22. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell PAMI-8:679–698

The RRDS, an Improved Animal Experimentation System for More Animal Welfare and More Accurate Results

Theo Gabloffsky, Alexander Hawlitschka, and Ralf Salomon

Abstract

Research of image recognition allows for improvements in animal welfare compliant and increase in data yield in animal experiments. One application for improvements are the so-called rotational tests with rats in Parkinson research. Here, the *Rat Rotation Detection System* (RRDS) frees the rat from the usually used breast belt while achieving similar results as the previous system, with a difference of 12.4 %. RRDS basically consists of an off-the-shelf camera combined with a YoloV4-Neural-Network, which detects the coordinates of the head, the tail, and the torso of the rat. With these coordinates, RRDS calculates two vectors, which are further used to calculate the rotation of the rat. The RRDS is a step towards improved animal welfare and more accurate results in animal experimentations.

Keywords

Rotation-detection · Animal-welfare · Object-detection · Parkinson-research rotometer

T. Gabloffsky (✉) · R. Salomon
University of Rostock, Rostock, Deutschland
e-mail: theo.gabloffsky@uni-rostock.de

R. Salomon
e-mail: ralf.salomon@uni-rostock.de

A. Hawlitschka
Institute of Anatomy, Rostock University Medical Center, Rostock, Deutschland
e-mail: alexander.hawlitschka@med.uni-rostock.de

V. Lohweg (Hrsg.), *Bildverarbeitung in der Automation*, Technologien für die intelligente Automation 17, https://doi.org/10.1007/978-3-662-66769-9_5

1 Introduction

Ongoing research of image recognition allows for improvements in animal welfare compliant and increase in data yield in animal experiments. One application for improvements are the so-called rotational tests in animal models of Parkinson research. In these tests, the goal is to determine the amount of the clockwise and counter-clockwise rotations of the rats. The state-of-the-art of these rotational tests consists of the following four components, which can be seen in Fig. 1: (a) a bowl in which the rat rotates, (b) a breast belt, which is attached to the rat, and (c) a stiff wire, which connects the breast belt to (d) a rotation counter, which determines the amount of rotations.

This setup has the following drawbacks:

1. Due to simplicity of the electronic device, the output of the rotary counters are only net numbers. At the end of the experiment, the examiner only knows how often the rat rotated clockwise and counter-clockwise. The system does not give any time-relevant information.
2. The system is not able to distinguish between rotations and any further movement, like standing up, rolling around or self-cleaning.
3. The breast belt may harm the movement of rat or manipulate its activity.

This paper introduces the *Rat Rotation Detection System*, or *RRDS* in short. This system is meant to replace the measurement setup of the rotary detectors. The RRDS works in four basic steps. (1) A camera records the rotation of the rat in the bucket. This video is passed onto a (2) Feature Detection, in this particular application a *YoloV4*-Network which identifies head, tail, and torso of the rat. For each frame, the output of the network are two-dimensional coordinates of the detected features. (3) With these coordinates of the head, the tail and the torso, the system generates two angles. (4) With the changes of these angles over time (frame by frame), the system determines the rotation of the rat. Sect. 4 describes the system in more detail.

RRDS was evaluated in a real-world experiment with laboratory animals. In the scope of the experiments, RRDS was compared with the state-of-the-art breast belt system. The laboratory setup is described in Sect. 5.



Fig. 1 The figure shows a rat in a test setup with a mounted breast belt

The results, as further described in Sect. 6, show that RRDS generates similar results as the breast belt system with a difference of 7.6 % for rats medicated with amphetamine, and 16.7 % for rats medicated with apomorphine. The deviation occurs because of the measuring setup as further described in Sect. 7, RRDS is a contribution to more animal welfare and an improvement of the grain of detail in experiments with laboratory animals.

2 Rotational Test in Parkinson Research

In Parkinson research, rats serve as experiment animals and models for this human disease. To generate Parkinson-like symptoms in rats, the animals underlay a surgery which destroys the dopamine-producing part of their brain (the substantia nigra pars compacta). In order to identify whether the surgery was successful or not, the rats undergo a so called *rotation test*. In this test, the rat gets an injection of either amphetamine or apomorphin. In the case of amphetamine and a successful surgery, the rats rotate clockwise. The application of apomorphin causes the rat to turn counterclockwise. The mean rotation speed within a defined period of time indicate the surgery's success. The state-of-the-art measurement setup for the rotational test is based upon the Brest belt system as described in Sect. 1. [1]

Short Comings of the Breast Belt System: Although the term about rotations can be found in works concerning this topic, the solution based on the breast belt and the rotary detectors mainly detect the movement of the upper body of the rat. If a rat moves its upper body, the rotary detector counts the movement of the wire as a rotation, even though the rat did not rotate at all. This may happen through extensive self-cleaning of the rat. Therefore, the conversion of the numbers presented by the rotary detectors into rotations seems to be problematic. In addition, the system does not present any time-dependent information about the rotations. In the end, only the numbers of clockwise rotations and of the counterclockwise rotations is known. For research on the medications, a further knowledge about the characteristics of the rotations is valuable, e.g., changes in rotation speed during the test or even brief changes in direction. These parameters can not be detected by the Brest belt system as it is. In addition, the breast-belt may inhibit the movement of the rat.

Previous Work: Mouse-Pi The system *Mouse-Pi*, stated in [2], tried to enhance the experiments with a video based approach as well. It mainly consists of a Raspberry Pi connected to a camera. In contrast to RRDS, the rats were marked with coloured blobs. The Raspberry Pi tried to detect the coordinates of the blobs. Out of these coordinates, it calculated the rotation of the rat. Although the system showed good results in basic testing with blobs on a sheet of paper, it failed in the field. The reasons for that are stated as: (1) The texture of the rats fur does not allow for a sufficiently homogenous coloring. (2) Solid blobs cannot be applied, because the rat may try to remove them from their body. This may harm their behaviour which would invalidate any experiment. (3) The

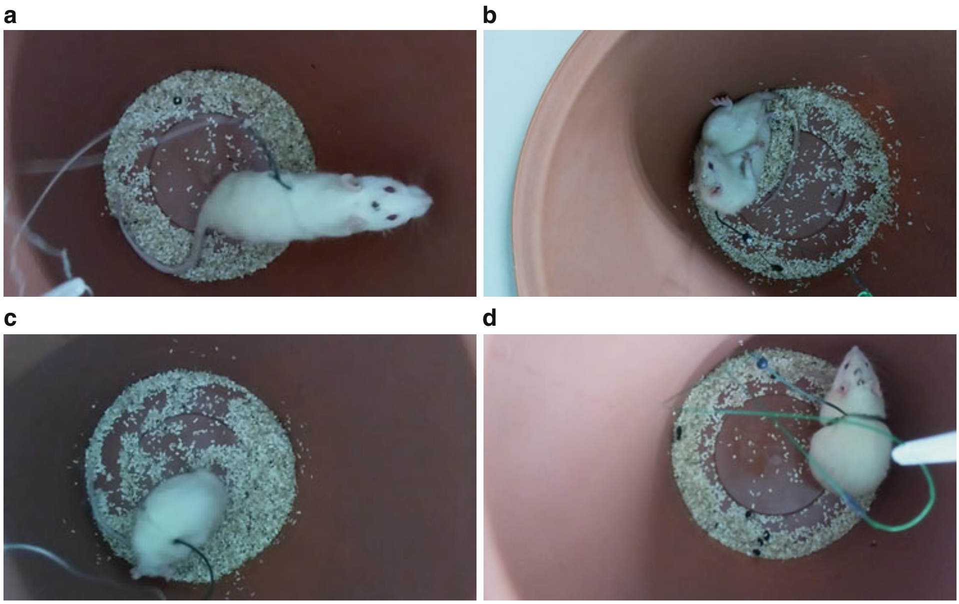


Fig. 2 The Figure shows different movements of the rat: (**a**) standing, (**b**) rolling, (**c**) a rat with ducked head and tail, (**d**) a rat which sits on its tail.

blobs are not visible for the Raspberry Pi all the time through the experiment because of the movement of the rat.

Fig. 2 shows different moving behaviours of the rats, including standing, rolling and self-cleaning. In addition, the rats sometimes tend to duck the head or the tail.

3 Similar Approaches

The idea behind an automated position estimation of laboratory animals is already discussed in various literature. Recent studies involve the detection of limbs and pose estimations of mice [3] or pigs [4] using an Opensource-Software *DeepLabCut* [5].

4 The Rat Rotation Detection System (RRDS)

This Section presents RRDS, an improved approach of refining the rotational test. Fig. 3 illustrates the general setup of RRDS which consists of the following components:

Video Camera: A video camera is monitoring the rat in its bucket. For the application at hand, a Raspberry Pi video camera with a decent resolution of 1280×720 pixels at 25 frames per second (fps) is entirely sufficient.

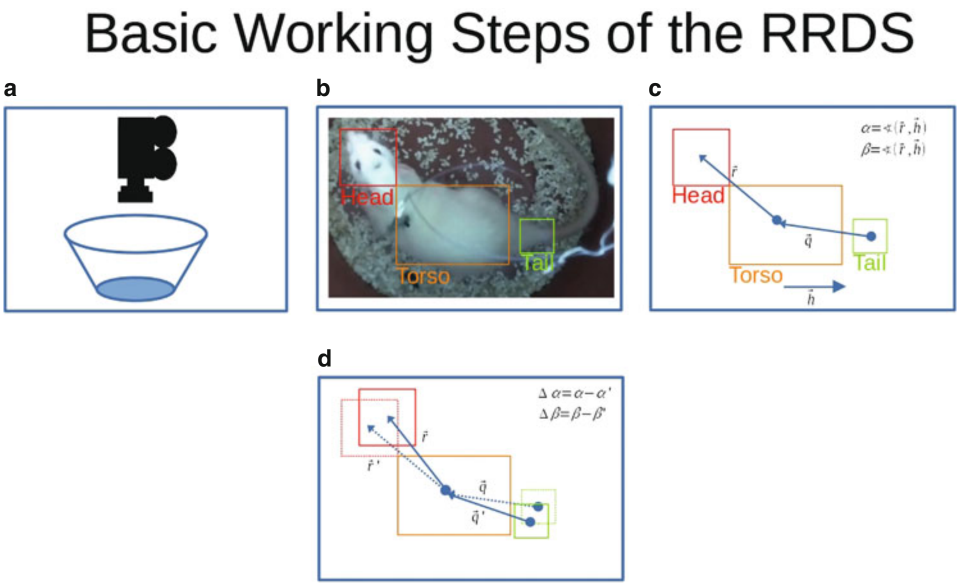


Fig. 3 The Figure illustrates some parts of the working principle of RRDS, which involves the following steps: (**a**) a video camera records the rat from above. (**b**) A neural network detects certain features: head, tail, and torso of the rat. (**c**) Out of these features, the system calculates two vectors: $\vec{r}$ from head to torso and $\vec{q}$ tail to torso. (**d**) With the change of the angles between the vectors and a help vector $\vec{h}$, the system concludes of the rotation of the rat.

Feature Extraction by a Neural Network: The first processing stage consists of the well known anchor-box based neural network *YoloV4* [6]. Though in comparison with other CNN-based networks, e.g., RetinaNet [7], the YoloV4-network achieves a better performance in the COCO-dataset [8]. In the scope of this application, the accuracy of the Yolo-network should be more than sufficient. Furthermore, the YoloV4-network achieves faster processing (20 fps faster). The task of the network is to extract bounding boxes of the following features of the rat: (1) the head, (2) the tail and (3) the torso. For this purpose, the network analyzes the video of the rat frame by frame. Though the output of the network are bounding boxes which include the (x, y) coordinates and the width and height of the detected object. For calculating the rotation of the rat, the relevant part are the following coordinates:

$$P_{\text{head}} = (x_{\text{head}}, y_{\text{head}})$$
$$P_{\text{tail}} = (x_{\text{tail}}, y_{\text{tail}})$$
$$P_{\text{torso}} = (x_{\text{torso}}, y_{\text{torso}})$$

Angle Calculation: the second processing stage takes the feature's coordinates and constructs the two vectors $\vec{r} = (r_x, r_y)^{\mathrm{T}}$ and $\vec{q} = (q_x, q_y)^{\mathrm{T}}$ out of them:

$$\vec{r} = P_{\text{torso}} - P_{\text{head}}$$
$$\vec{q} = P_{\text{torso}} - P_{\text{tail}}$$

In order to detect changes in the position of the rat, e.g., through rotation, the system calculates the angles α and β between the vectors $\vec{r}$, $\vec{q}$ and a third, constant vector $\vec{h}$. In general, the angle between two vectors $\vec{r}$ and $\vec{h}$ is calculated in the following way:

$$\cos(\alpha) = \frac{\vec{r} \cdot \vec{h}}{|\vec{r}| \times |\vec{h}|} \tag{1}$$

Through setting the vector $\vec{h}$ to $\vec{h} = (1, 0)^{\mathrm{T}}$, the equations simplifies to:

$$\alpha = \operatorname{acos}\left(\frac{r_x}{\sqrt{r_x^2 + r_y^2}}\right) \tag{2}$$

$$\beta = \operatorname{acos}\left(\frac{q_x}{\sqrt{q_x^2 + q_y^2}}\right) \tag{3}$$

Calculating the Rotation: The calculated angles α_t and β_t of the analyzed frame at the time t are subtracted from the angles of the previous frame: $\Delta\alpha = \alpha_t - \alpha_{t-1}$ and $\Delta\beta = \beta_t - \beta_{t-1}$. Both angles are summed up from frame to frame to $\alpha_s = \sum \Delta\alpha_f$ and $\beta_s = \sum \Delta\beta_f$. If both of these sums are greater than a specified angle-accuracy, e.g., $\theta = 6$ deg, the clockwise-rotational counter C_c is increased and the sums are reduced by the angle-accuracy: $\alpha_s = a_s - \theta$, $\beta_s = \beta_s - \theta$. If both of the sums are smaller than the negative angle-accuracy, the counter-clockwise-rotational counter C_{cc} is increased and the sums are increased by the angle-accuracy: $\alpha_s = \alpha_s + \theta$, $\beta_s = \beta_s + \theta$. This approach imitates the working principle of the rotational counters. In addition, the changes of the values of the counters C_c and C_{cc} are used to calculated the rotating speed of the animal in [deg s^{-1}].

5 Prototypical Experiment with RRDS

This section describes the experimental setup in terms of the procedure of the rotational test as well as the measuring setup of the prototypical RRDS. The laboratory environment allows a measurement of four rats simultaneous. Therefore, four instances of RRDS camera systems took videos of the rats. In total, the examination included 18 rats which were treated with both medications on different days as mentioned in Sect. 2. Some of

a b

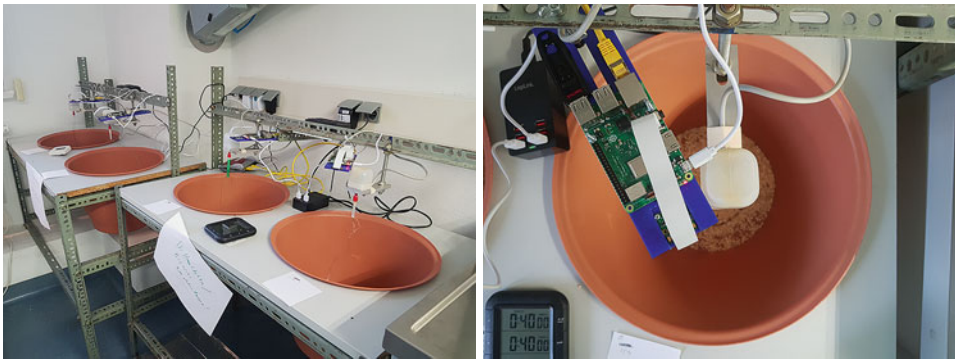


Fig. 4 (**a**) Experimental Setup of the medical research. (**b**) The Figure shows a camera of RRDS mounted next to the rotational counter

the videos were not recorded properly, which results in a total of 15 videos of rats treated amphetamine and 16 videos of rats treated with apomorphine.

Procedure of Rotational Test As seen in Fig. 4a, the experimental setup consists of four cylindrical buckets of approximately 30 cm in diameter. Inside of each bucket, a rat is rotating with a breast belt, as seen in Fig. 1. which is connected to an electromechanical rotary detector. Each rat has to be medicated by the examiner before the experiment can start. After the medication, each rat is getting mounted with the breast belt which is connected to the rotary detector. After a short wake-up time, the examiner resets the rotary detector and starts the experiment, which lasts typically 40 min for apomorphine and 60 min for amphetamine. During the experiments, the experimental laboratory is illuminated with dimmed light.

Prototypical RRDS in a Laboratory Environment Fig. 4b shows one instance of RRDS camera-system. The camera of this instance is mounted next to a rotational counter of the old measurement system. The mounting system is a special 3D printed holder. The camera-system contains a Raspberry Pi Camera V2.1 that is connected to a Raspberry Pi 3B+ (RPi). The RPi controls the camera in regard to starting, stopping and converting the video into h264-stream with 1280×720 px with 25 fps. In addition, the RPI transfers the video stream onto a mass storage device, which is connected over the USB 3.0 Port of the RPi. The RPis are controlled via Ethernet from a Laptop.

Feature Extraction After the RPi has stopped the recording of the videos, the data were transferred to an AI-capable computer with a NVIDIA RTX 2080TI GPU. The videos were analysed frame by frame with anchor-box based neural network *YoloV4* utilized by the DarkNet-Framework [9]. The network was trained with roughly 30.000 labeled images in order to learn to detect the features (1) head, (2) tail and (3) torso. Though the torso is the easiest feature to detect, the task to detect the tail is quite complicated. In order to fit into the RAM of the GPU (11 GB), the input resolution of the network was set

to 1024×672. The Batch-Size is set to 64 The initial learning rate was set to 0.01×10^{-3} in order to prevent a gradient explosion.

Resolution of Rotation-Detection: In order to compare both systems, the resolution for the rotation-detection of RRDS is set to 6 deg, which is equal to the resolution of the hardware of the rotary detector. However, smaller resolutions may generate more false detections of generations due to noise in the feature detection.

Limitations of the Experiment The comparison of both systems requires a simultaneous measurement with both systems of each rat. Therefore, the camera based system RRDS has to detect the features of the rat even though the visual field is impaired by the wire and the breast belt. The consequence of this is, that the training images taken to train the neural network are probably only suitable for this particular case. When using RRDS without the rotary detector, the neural network of the feature detection may need to train with new images.

6 Results of Experiments

This Section presents the results of the experiments described in Sect. 5. Though the reasons for different characteristics in the behaviour of the rats are quite interesting, the main focus lies on the evaluation of RRDS in comparison with the rotation encoder.

6.1 Rotation-Detection

The following tables (Table 1) present the difference $D = C_c - C_{cc}$ between the clockwise and counter-clockwise counters (C_c and C_{cc}) of the Brest belt system and the new RRDS.

Detection of Head, Torso and Tail In sum, the network was not able to detect the features in the following amount of frames:

- E_{head} 69 434
- E_{tail} 114 330
- E_{torso} 13 138

6.2 Rotation over Time

Fig. 5 shows the rotation over time of two different rats medicated with Amphetamine (Fig. 5b, d) and apomorphine (Fig. 5a, c). Both rats show different rotating characteristic, rotating clockwise and counter-clockwise. As Fig. 5d shows a more or less linear increase of both directions of rotation, the Fig. 5b shows a greater increase in the clockwise rotation.

Table 1 The table presents the difference between the net numbers given by the rotation encoder D_{BB} for rats rotating medicated with amphetamine (**a**) and apomorphine (**b**) as well as the difference between the counters measured by RRDS D_{RRDS}. The average deviation results in 7.5% for amphetamine and 16.7 % for apomorphine.

Rat	D_{BB}	D_{RRDS}	Diff. in %
Amp-1	–22 630	–22 611	0.8
Amp-2	–12 355	–7,801	36.8
Amp-3	–3 516	–3 005	14.5
Amp-4	–6 973	–7 639	9.5
Amp-5	–42 155	–40 931	2.9
Amp-6	–14 655	–14 550	0.7
Amp-7	3 484	2 733	21.5
Amp-8	–29 650	–28 508	3.8
Amp-9	–1 225	–1 186	3.11
Amp-10	–12 390	–12 207	1.47
Amp-11	–3 006	–3 002	0.1
Amp-12	–1 958	–1 960	0.1
Amp-13	–3 601	–3 600	0.0
Amp-14	8 873	8 283	6.2
Amp-15	–1 628	–1 844	13.5

Rat	D_{BB}	D_{RRDS}	Diff. in %
Apo-1	11 383	7 750	33.4
Apo-2	19 835	14 845	25.1
Apo-3	–212	–176	16.8
Apo-4	2 607	2 585	0.8
Apo-5	21 940	21 045	4.0
Apo-6	–292	–2 890	1.0
Apo-7	24 503	23 131	5.6
Apo-8	2 651	2 673	0.8
Apo-9	17 554	16 825	4.1
Apo-10	–3 747	–3 780	0.8
Apo-11	–366	–352	3.7
Apo-12	392	169	56.8
Apo-13	–328	–334	1.9
Apo-14	–507	–504	0.5
Apo-15	100	110	10.8
Apo-16	–93	–186	100.6

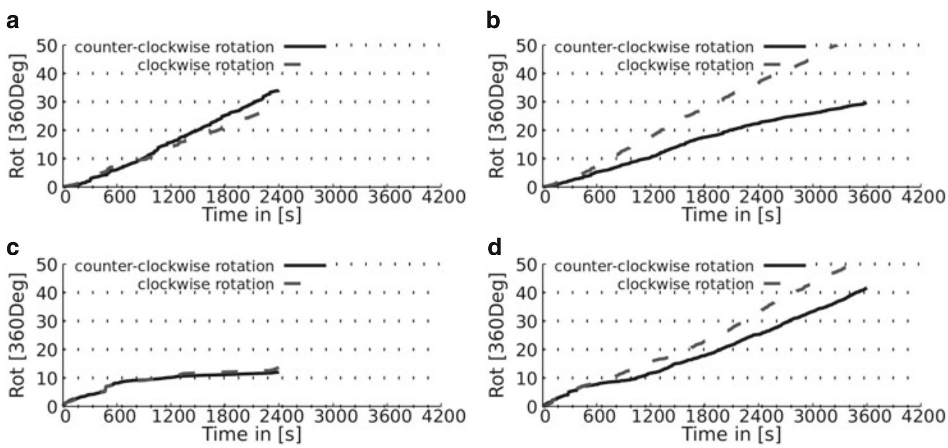


Fig. 5 The Figure shows the summarized changes over time of two rats medicated with apomorphine ((**a**) and (**c**)) and amphetamine ((**b**) and (**d**)). It shows, that different rats do not show the same response to the medication.

7 Discussion and Conclusion

Rotation Numbers: The experiment has shown that RRDS is able to detect similar numbers of rotation as the rotary detector with the breast belt. The results presented in Table 1 shows an average deviation of 7.6 % for the rats treated with amphetamine and 16.7 % for the rats treated with apomorphine. The average deviation of all rats is 12.1 %. Though the deviation seems to be large, it is neglectable. The reason for this is that the ratio between the difference of rotation and the sum of the rotations is quite small.

The main reason for the deviation lies in the detection of the head, tail, and torso. The neural network of RRDS is easily trainable to detect the features on a calm rat, as seen in Fig. 1. However, the difficulties in detection occur when the rat rotates with the head ducked and the tail under its body (as seen in Fig. 2c) or close to the side of its body. In addition, fast movements of the rat can result in blurred images (Fig. 6a), which are harder to detect for the neural network. In addition, the positioning of the camera was not optimal. Some of the rats tend to stand up and move their heads out of the scope of the camera as Fig. 6b shows. In addition, the wire of the breast belt and the rotary detector blocked a part of the vision of the camera onto the rat (Fig. 6c). This leads to more not-detectable frames in the video and furthermore to not-detectable rotations.

Rotation Over Time: Fig. 5 shows, that two rats show a different rotating characteristic even though both were medicated in the same way. These rotating characteristic will be addressed in future Parkinson research.

Drawbacks of RRDS: A drawback of RRDS lies in the overall energy consumption and hardware prerequisite. Due to the usage of the neural-network on a state-of-the-art GPU, the energy consumption can easily reach up to 250 W, which is significantly higher

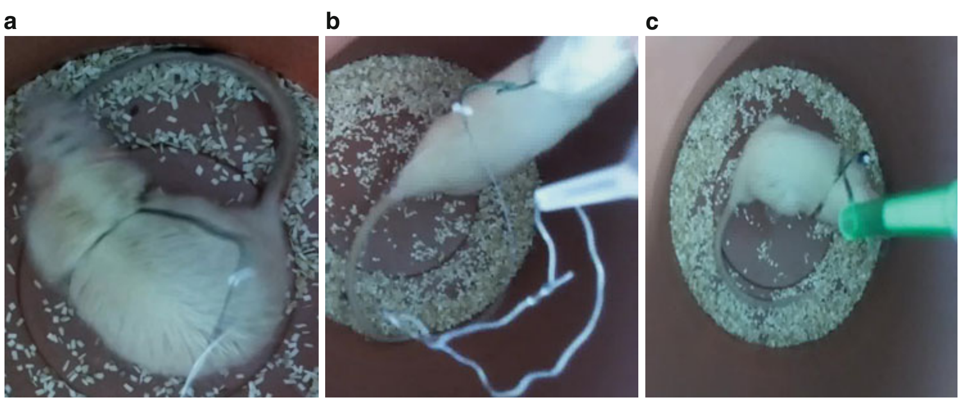


Fig. 6 The Figure shows different different situations in the videos, which lead to the deviation between the RRDS and the Brest belt system. (**a**) shows a blurred image, which can occur when the rat is moving fast. (**b**) shows a standing rat with its head out of the vision of the camera. (**c**) shows a frame of the video with a blocked view on the head of a rat.

than the consumption of the rotary detectors. In contrast, it delivers far more information about the rats than the rotary detectors.

Future Work: Based on the problems described above, future work will have the following goals:

1. In order to decrease the energy consumption, the feature detection of RRDS could be replaced by a smaller neural network or a more energy-efficient algorithm.
2. RRDS is able to generate more data from the experiments. Future Parkinson research will address these information which may lead to a better understanding of the disease.
3. Testing the accuracy of other networks, e.g., DeepLabCut, with the generated training data.
4. Adapting the network in order to generate a behaviour profile of the rat, including standing, rolling and self-cleaning.
5. Though the state of RRDS is more or less prototypical, the examiner needs fundamental knowledge about the workflow inside the system. Future work will target the automatization of the working processes, with the aim of a user friendly system.

In conclusion, RRDS is a step to more animal welfare and more accurate results in animal experimentations.

Acknowledgement The authors gratefully thank Prof. Wree for pointing the authors to this problem and performing the surgeries on the rats. The authors also thank the two teams led by Prof. Wree and Prof. Kipp for providing the laboratory equipment, and their support during the experiments.

References

1. Antipova V, Hawlitschka A, Mix E, Schmitt O, Dräger D, Benecke R, Wree A (2013) Behavioral and structural effects of unilateral intrastriatal injections of botulinum neurotoxin a in the rat model of Parkinson’s disease. J Neurosci Res 91(6):838–847
2. Joost R, Ziese D, Hawlitschka A, Salomon R (2018) Mouse-pi: A platform for monitoring in-situ experiments. In: Jasperneite J, Lohweg V (Hrsg) Kommunikation und Bildverarbeitung in der Automation. Springer, Berlin, Heidelberg, S 246–257
3. Weber RZ, Mulders G, Kaiser J, Tackenberg C, Rust R (2022) Deep learning-based behavioral profiling of rodent stroke recovery. BMC Biol 20(1):232. https://doi.org/10.1186/s12915-022-01434-9
4. Gorssen W, Winters C, Meyermans R, D’Hooge R, Janssens S, Buys N (2022) Estimating genetics of body dimensions and activity levels in pigs using automated pose estimation. Sci Rep 12(1):15384. https://doi.org/10.1038/s41598-022-19721-4
5. Mathis A, Mamidanna P, Cury KM, Abe T, Murthy VN, Mathis MW, Bethge M (2018) Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. Nat Neurosci 21(9):1281–1289. https://doi.org/10.1038/s41593-018-0209-y
6. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: Optimal speed and accuracy of object detection https://doi.org/10.48550/ARXIV.2004.10934
7. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection https://doi.org/10.48550/ARXIV.1708.02002

8. Lin T, Maire M, Belongie SJ, Bourdev LD, Girshick RB, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. CoRR. http://arxiv.org/abs/1405.0312
9. Bochkovskiy A https://github.com/artynet/darknet-alexeyab

A Study on Data Augmentation Techniques for Visual Defect Detection in Manufacturing

Lars Leyendecker, Shobhit Agarwal, Thorben Werner, Maximilian Motz, and Robert H. Schmitt

Abstract

Deep learning-based defect detection is rapidly gaining importance for automating visual quality control tasks in industrial applications. However, due to usually low rejection rates in manufacturing processes, industrial defect detection datasets are inherent to three severe data challenges: data sparsity, data imbalance, and data shift. Because the acquisition of defect data is highly cost-intensive, and Deep Learning (DL) algorithms require a sufficiently large amount of data, we are investigating how to solve these challenges using data oversampling and data augmentation (DA) techniques. Given the problem of binary defect detection, we present a novel experimental procedure for analyzing the impact of different DA-techniques. Accordingly, preselected DA-techniques are used to generate experiments across multiple datasets and DL models. For each defect detection use-case, we configure a set of random DA-pipelines to generate datasets of different characteristics. To investigate the impact

L. Leyendecker (✉) · S. Agarwal · M. Motz
Fraunhofer Institute for Production Technology IPT, Aachen, Deutschland
e-mail: lars.leyendecker@ipt.fraunhofer.de

S. Agarwal
e-mail: shobhit.agarwal@ipt.fraunhofer.de

M. Motz
e-mail: maximilian.motz@ipt.fraunhofer.de

T. Werner
Information Systems and Machine Learning Lab (ISMLL), Hildesheim, Deutschland
e-mail: werner@ismll.de

R. H. Schmitt
Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen, Aachen, Deutschland
e-mail: r.schmitt@wzl-mq.rwth-aachen.de

V. Lohweg (Hrsg.), *Bildverarbeitung in der Automation*, Technologien für die intelligente Automation 17, https://doi.org/10.1007/978-3-662-66769-9_6

of DA-techniques on defect detection performance, we then train convolutional neural networks with two different but fixed architectures and hyperparameter sets. To quantify and evaluate the generalizability, we compute the distances between dataset derivatives to determine the degree of domain shift. The results show that we can precisely analyze the influences of individual DA-methods, thus laying the foundation for establishing a mapping between dataset properties and DA-induced performance enhancement aiming for enhancing DL development. We show that there is no one-fits all solution, but that within the categories of geometrical and color augmentations, certain DA-methods outperform others.

Keywords

Deep learning · Defect detection · Data augmentation · Manufacturing · Visual quality control

1 Introduction

Manufacturing processes have been optimized in recent decades to achieve minimum reject rates and high product qualities. However, as product and process complexities increase, the importance of reliable quality continues to grow. Defects such as internal holes, pits, abrasions, and scratches on workpieces or knots, broken picks, and broken yarn in fabrics [1] negatively impact both visual and functional product properties [2]. Defects also contribute to the additional wastage of resources, safety hazards, and can have severe economic consequences for a company. Therefore, reliably assuring the quality of manufactured products is of paramount importance in manufacturing. One of the famous and contemporary solutions towards achieving the goal of a fully automated quality control system is through deep learning (DL)-based computer vision. DL algorithms improve over existing rule-based systems in terms of generalization and performance, while requiring less domain expertise [3–5]. However, a major disadvantage of data-driven approaches compared to rule-based techniques lies in the strong dependency of model precision on data quantity, data quality, and the evolution of the data over time (data drift) [6]. While the focus in recent years has been on the development of advanced network architectures (e.g., ResNet-50 [7] or Inception-v3 [8]), the progress that is being made in model-space is increasingly diminishing. As a result, the development is shifting more towards data-centric approaches, especially in real-world domains like for example manufacturing or medical diagnostics. Table 1 provides an overview of the main data challenges that are characteristic for image data acquired from production processes. These properties form a strong contrast to the ones of (research) datasets (e.g., ImageNet [9], COCO [10], MNIST [11]) used for developing and benchmarking of deep neural network architectures and DL-algorithms, which is why the approaches from research are difficult to transfer one-to-one to such complex defect detection use-cases.

Table 1 Causes of data quality issues in DL-based visual defect detection in terms of data sparsity, data imbalance and data shift

Data Quality Issue	Description
Amount of data	Difficulty in collecting sufficiently large amounts of data.
Label inconsistencies	Labor-intensive task that is oftentimes ambiguous and usually requires multiple domain-experts
Data imbalance	Defective parts tend to be significantly underrepresented compared to non-defective ones
Changing lightning conditions	Contrasts and brightness changes across different work shifts
Exposure issues	Reflections and shadows cast by complex components
Sensor failure	Image failures or high noise-levels due to sensor degradation amplified by harsh environments
Changing object poses	Especially in mass production often different orientation of components
Changing appearances	Changes in the appearance of a product from time to time can make the data previously collected unusable.

Data augmentation (DA) represents a data-space solution addressing the above mentioned data quality challenges. There are various DA techniques that aim for changing both the geometrical and visual appearance of images to improve both performance and robustness properties of deep neural networks. The most common DA techniques are geometric transformations, color augmentations, kernel filters, mixing images and random erasing [12]. Even though DA is already an integral part of DL pipelines, different DA-methods are often blindly applied based on empirical knowledge and require elaborate tuning for specific datasets. To analyze the impact of different DA-methods on both precision and generalization for the task of visual defect detection, this paper introduces our experimental procedure in Sect. 3.3, presents the results in Sect. 4.2 and finally derives insights about the studied DA-methods in Sect. 5. Sect. 3.2 introduces the three real-world datasets which we work with. Our DA-methods are chosen according to a preliminary study of related papers that is summarized in Sects. 2 and 3.3.

2 Related Work

This section provides a brief overview of work that addresses the generalization problem, DA approaches, and its impact on real-world DL tasks. One central drawback of real-world datasets is that the models trained on them do not generalize well as these datasets are prone to domain shift [13]. In recent years model-centric techniques such as dropout [14], transfer learning [15], and pretraining [16] have tried to address the issues of generalization, particularly in deep neural networks. DA tries to avoid poor generalization by solving the root problem of training data [17] rather than changing the model or training process. Applications of DA can be found in various works across multiple

domains such as natural language processing [18], computer vision [17], and time series classification [19]. Particularly in computer vision tasks DA has been applied to address the domain generalization problem [20–22]. Many papers exist that apply and analyze basic DA-techniques (e.g., oversampling and data warping on histopathological images [23]) and advanced methods (e.g., stacked DA on medical images [24], style-transfer augmentations [25], cGan, and geometric transformations [26]) for specific use cases and datasets.

Fewer papers exist that provide an overview of DA-methods and try to examine their influences on model accuracy. The survey of Shorten et al. [17] presents a comprehensive overview of DA and present the impact examination of individual methods on well-known datasets (e.g., CIFAR-10, MNIST, Caltech101) in an isolated manner of pairwise comparisons. Shijie et al. [27] explore the impact of various DA-methods on image classification tasks with CNNs. On subsets of CIFAR10 and ImageNet, they conduct pair and triple comparisons to identify best-performing DA-techniques and to draw general conclusions. Yang et al. [28] systematically review different DA-methods and propose a taxonomy of reviewed methods. For semantic segmentation, image classification, and object detection, they compare the performances of different model architectures on datasets (e.g., CIFAR-100, SVHN) with and without pre-defined set of DA-techniques. The survey paper of Khosla et al. [29] presents an overview of selected DA-methods without conducting further effect analyses. In addition to generic studies on scientific datasets, a few domain-specific approaches exist. The only related work on DA in defect detection is provided by Jain et al. [30]. They propose a DA-framework utilizing GANs which they use to investigate data synthetization for classification of manufacturing datasets.

Scientific Impact

Existing studies are almost exclusively conducted on scientific datasets and no reference is made to specific application domains (with the exception of [30]). To the best of our knowledge, there is currently no preliminary work, that examines the impact of DA-methods specific to DL-based visual quality control in manufacturing datasets in an unconstrained setting (i.e. only pairwise evaluations).

3 Approach

In this section, we present our approaches and procedures. Sect. 3.1 defines the mathematical problem of binary defect detection. Sect. 3.2 introduces the datasets considered in this study and their properties. The experimental procedure, the domain shift measure, and the evaluation metrics are presented in Sect. 3.3.

3.1 Binary Defect Detection Problem Definition

For binary visual defect detection, the input feature space is denoted by $\mathcal{X}$ and $\mathcal{Y}$ denotes the target space. We define the domain as a joint distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$ and the dataset as $\mathcal{D} = \{(\mathcal{X}_i, \mathcal{Y}_i)\}_i^N$, where N is the number of training examples. In this work, $\mathcal{X}^1$, $\mathcal{X}^2$, $\mathcal{X}^3$ comprises images from three datasets, namely: (1) AITEX fabric defects [1], (2) Magnetic tile defects [31], and (3) TIG Aluminium 5083 welding defects [32]. We define the binary classification problem where $\mathcal{Y} \in$ {Defected, Non-defected}. Furthermore, the DL model is defined as $f : \mathcal{X} \to \mathcal{Y}$, where the primary objective is to learn a mapping from the input space $\mathcal{X}$ to target space $\mathcal{Y}$. In this work $f \in$ {ResNet-50 [7], Inception-V3 [8]}. The predictions generated using model f are denoted as $\hat{\mathcal{Y}}$. The categorical cross entropy loss function is defined as $\ell : \mathcal{Y} \times \hat{\mathcal{Y}} \to [0, \infty)$. Each dataset $\mathcal{D} = \{(\mathcal{X}_i, \mathcal{Y}_i)\}_i^N$ is augmented using various DAs, where θ denotes the list of all DAs, and a new augmented dataset is generated as $\mathcal{D}^1 = \theta(\mathcal{D})$. For each dataset, ten DA-pipelines with varying DAs are constructed to create ten different data sets $\mathcal{D}^1$ .. $\mathcal{D}^{10}$.

3.2 Presentation of the Datasets

Three real-world industrial-grade datasets are used in this work. An overview of exemplary images is provided in Fig. 1. The Magnetic tile defects dataset (MagTile) contains a total of 1,344 images of magnetic tiles with five defect types: blowhole, crack, fray, break, nneven (grinding uneven), and free (no defects). AITEX is a fabric production dataset containing 246 images of 4,096 × 256 pixels that capture seven different fabric structures. In total, there are 140 defect-free images, 20 for each type of fabric, and there are a total of 105 images with defects. The TIG Aluminium 5083 welding seam dataset (TIG5083) contains 33,254 images of aluminium weld seams and the surrounding area of the weld seam, with six classes: good weld, burn through, contamination, lack of fusion, misalignment, and lack of penetration. We convert the multi-class classification task of all datasets into a binary classification problem by merging all individual defect types into a single defect class.

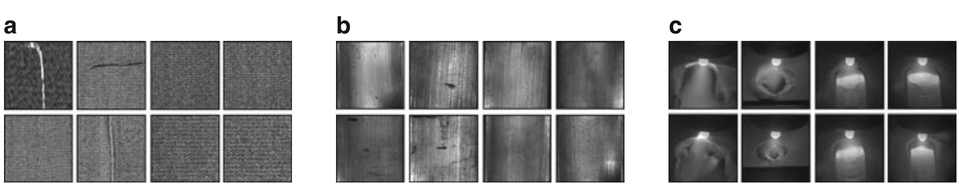


Fig. 1 Exemplary raw images of the datasets studied: AITEX (**a**), MagTile (**b**), and TIG5083 (**c**)

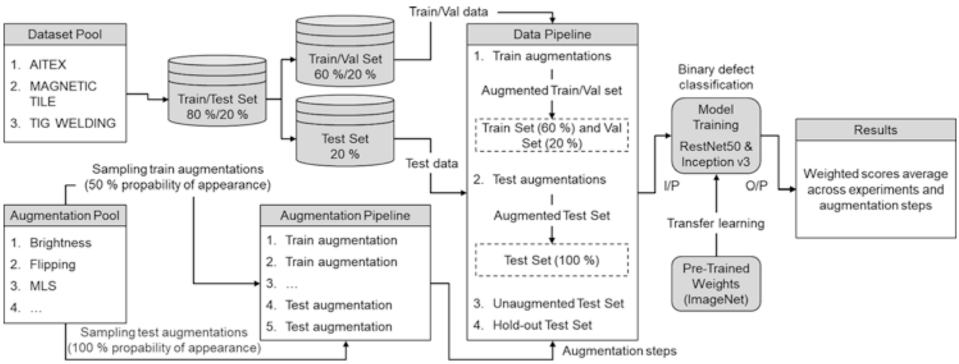


Fig. 2 Experiment protocol for constructing the DA-pipelines, training, and evaluating the defect detection model

3.3 Experiment Procedure

To evaluate the impact of DA-techniques we propose a three-stage process: First, for each dataset, apply a DA-pipeline and evaluate model performance on different test sets. Second, measure the domain shift between the train set and the test sets. Third, correlate the achieved performance with the domain shift. This framework provides insight into the effects of different DAs on model performance, domain shift, and, through the correlation of both, the generalization capabilities of the trained model. An overview of our algorithm can be found in Fig. 2. We assume a standard train-test split of 80/20 and further a validation split of 60/20 (based on the 80% train split). Additionally, we create a hold-out test set by splitting off one of the defect classes per dataset before they are merged (see Sect. 3.2). This hold-out set serves as an additional out-of-distribution test set to measure the generalization capabilities of the model. We apply DA in two different settings. For AITEX and MagTile, augmented datapoints were added as new instances, retaining the original ones. This was done to increase the overall number of instances in the dataset and stabilize training. For TIG5083, augmented datapoints replace the originals since the dataset already contains enough images for training. The hold-out class for the AITEX data set was ‘Broken end’, the hold-out class for Magnetic tile defects was ‘Crack’, and the hold-out class for TIG Aluminium 5083 was ‘contamination’ class.

Data Augmentation Pipelines

In order to pre-select the DA-steps for this paper, a survey was conducted across 24 papers dealing with 6 major industrial image data sets. Table 2 describes all available augmentations for each dataset. From these augmentation pools, different pipelines for each dataset were constructed. For each pipeline, two of the augmentations are reserved for the test set and are later referred to as test augmentations. The remaining DAs have a 0.5 chance of being applied to the training set. This process is repeated ten times (see Table 3).

Table 2 Preselected set of DA-methods for TIG5083, AITEX, and MagTile

TIG5083	AITEX	MagTile
1. Gaussian Noise	1. Gaussian Noise	1. Salt & Pepper Noise
2. Transpose Image	2. Transpose Image	2. Transpose Image
3. Flip Image	3. Flip Image	3. Flip Image
4. Perspective Transformation	4. Random Perspective	4. Random Perspective
5. Add Brightness	5. Color Jitter	5. Color Jitter
6. Affine Transformation	6. Moving Least Squares (MLS)	6. MLS [33]
	7. Random Erase [12]	7. Retinex[34, 35]
	8. Random Rotate	

Table 3 Train, validation, and test set DA-Pipelines (AITEX)

Nr.	Train & Validation augmentations	Test augmentations
1	Random Perspective, Flip Image, Color Jitter	Random Rotate, Transpose Image
2	Flip Image, Random Perspective	Transpose Image, Random Erase
3	Gaussian Noise, Color Jitter, Random Perspective, Random Rotate, Flip Image	Random Erase, MLS
4	Random Erase, MLS, Gaussian Noise	Color Jitter, Random Perspective
5	Random Rotate, MLS, Gaussian Noise	Transpose Image, Flip Image
6	Random Perspective, Random Rotate	Gaussian Noise, MLS
7	Random Erase, AddNoise, Color Jitter	Random Rotate, Random Perspective
8	Random Rotate, Random Erase, Flip Image, Color Jitter	Random Perspective, Gaussian Noise
9	Color Jitter, Random Rotate, Gaussian Noise	MLS, Random Perspective
10	Random Perspective	Random Rotate, MLS

App. 6.1 provides an overview of selected unaugmented and augmented images for all three datasets.

Domain Shift Measures

We use an algorithm proposed by [36] for measuring the domain shift between datasets. In computer vision tasks, calculating domain shift can be seen as calculating the difference in representation by a model given the source and target domain. Given that a source domain is distant from the target domain, the representation of the domains in the learned space for a specific model tends to diverge. The authors used the activation values from the model's last layers to quantify the domain shift. Specifically, by creating a statistical distribution using each kernel's activation value in those layers, we can measure the distance between the datasets using the Wasserstein distance.

Evaluation Metrics

To evaluate the results of the binary classification problem, various metrics such as F1-Score, precision, recall, Jaccard similarity [37], Cohen's kappa score [38], and Matthews correlation coefficient (MCC) [39] are used. Since the datasets are imbalanced even after applying DA, all metrics (Jaccard, precision, recall, and F1-Score) are weighted by the class distribution. We use multiple different evaluation metrics, as they all slightly deviate from each other. In this way, we circumvent the difficulties due to the sensitivity of individual metrics and obtain a more conclusive evaluation. Since all these scores are bound between [0, 1] we average all of them for our reporting of final performance values.

4 Results

In this section, we present the results. Sect. 4.1 defines the training and implementation procedure. Sect. 4.2 provides an overview of the protocol followed to evaluate the results at the example of the AITEX dataset. Sect. 4.3 presents the results of our ablation study.

4.1 Training and Implementation

For controlling the model training, a validation set is split of from the augmented training set. The model is evaluated on the original test set, augmented test sets (using the two reserved test augmentations) and the hold-out set as described in Sect. 3.3. The hold-out class for the AITEX data set was 'Broken end', the hold-out class for MagTile defects was 'Crack', and the hold-out class for TIG5083 was 'contamination'. As models for our experiment, ResNet-50 and Inception-v3 were chosen, as both are widely used in the literature about industrial applications. The learning rate for both models is set to 10^{-3}, the Adam optimizer [40] is used and the first-layer input shape of the networks is set to 224 and 299 respectively. We initialize the networks using pre-trained weights (ImageNet) for both architectures. DL is enhanced via transfer learning with 50 epochs of frozen weights in the encoder (shallow training) and additional 30 epochs of fine-tuning the entire model (deep training). Similarly to the evaluation metrics, the class-balanced version of the loss function was employed to stabilize the learning process. The data for each experiment was normalized according to the statistics of the train set after applying DA.

4.2 Results for the AITEX Dataset

Fig. 3 depicts the average F1-Score across both the models and across the DA steps for each test set. The values are obtained by averaging the performance of each pipeline that contains the respective augmentation. We observe that the performance on the original test and, to a lesser extent, the augmented test set remains stable, but on the hold-out set

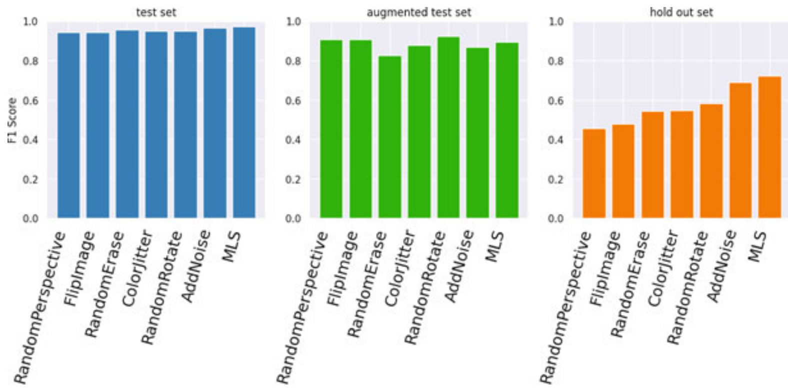


Fig. 3 F1-Score averaged across models for each DA-method (AITEX). Sorted by hold-out performance.

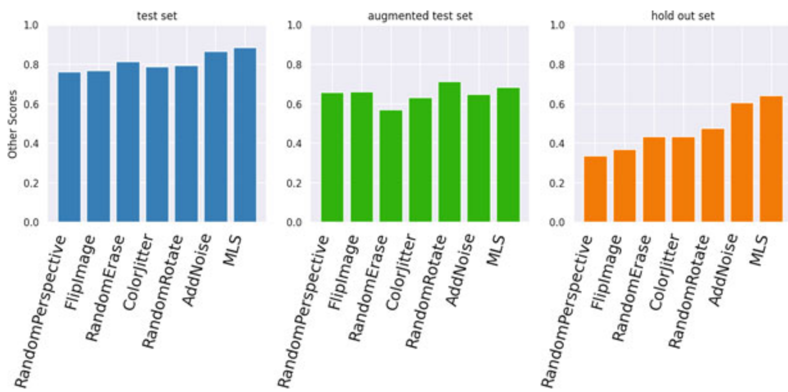


Fig. 4 Averaged Jaccard, precision, recall, kappa and MCC scores across models for each DA-method (AITEX). Sorted by hold-out performance.

(highest amount of domain shift) model performance significantly improved. The top three DA-steps for AITEX dataset are MLS, Gaussian noise and random rotating. As stated in Sect. 3.3, we also averaged the performance across multiple other metrics, since they all slightly differ from each other. Similar trends can be observed in Fig. 4.

Next, the distance between the train set (source domain) and the test sets (target domain) was calculated for all the models and datasets. Table 4 contains the mean and standard deviation across all the pipelines for the AITEX dataset and ResNet-50 model.

Table 4 Domain shift measure averaged across DA-pipelines for the last layer of ResNet-50

Train/Test	Train/Aug_test	Train/Hold_out
0.0764 ± 0.0831	0.0808 ± 0.0804	0.1841 ± 0.1981

Table 5 Pearson correlations between the domain shift and model F1-Scores (AITEX). The bold values represent the largest negative mean correlations value.

Pipe-line	Inception v3			ResNet-50			Mean
	Last layer	2nd Last layer	3rd Last layer	Last layer	2nd Last layer	3rd Last layer	
1	–0.996	–0.996	–0.998	–0.999	–0.999	–0.999	–0.998 ± 0.002
2	–0.727	–0.734	–0.505	–0.941	–0.940	–0.979	–0.804 ± 0.167
3	–0.989	–0.990	–0.971	–0.960	–0.967	–0.986	–0.977 ± 0.012
4	–0.970	–0.972	–0.995	–0.352	–0.317	–0.530	**–0.689 ± 0.297**
5	–0.916	–0.916	–0.986	–0.900	–0.905	–0.979	–0.934 ± 0.035
6	–1.000	–1.000	–1.000	–0.917	–0.931	–0.996	–0.974 ± 0.036
7	–0.999	–0.996	–0.988	–0.935	–0.946	–0.826	–0.948 ± 0.060
8	–0.999	–0.998	–1.000	–0.955	–0.916	–0.974	–0.974 ± 0.0307
9	–0.952	–0.955	–1.000	–0.341	0.121	–0.785	**–0.652 ± 0.411**
10	–0.994	–0.994	–0.991	–0.989	–0.987	–0.994	–0.991 ± 0.003
	–0.954 ± 0.084	–0.955 ± 0.082	–0.943 ± 0.154	–0.829 ± 0.256	–0.779 ± 0.375	–0.905 ± 0.152	

The domain shift increases from the original test set to the augmented test set to the hold-out set. Finally, the domain shift is correlated to the respective F1-Scores, as Wasserstein distance alone lacks interpretability.

A negative correlation means that with increasing domain shift the performance of the model on the test data decreases. Therefore, a greater correlation is desirable. Each cell in Table 5 contains the Pearson correlations between the distance measure and F1-Scores across all the test sets. Since the domain shift is measured based on a single layer of the model we evaluated the last three layers of each model and reported the values separately in the columns. The correlation values don't change depending on the layer used, but we observe two outliers in the pipelines that display a weaker correlation between domain shift and model performance. Further information can be found in App. 6.2. The same evaluation protocol was followed for evaluating the results across the other two datasets as well and similar trends were observed. The results TIG5083 and MagTile can be found in App. 6.3.

4.3 Results of the Ablation Study

In addition to the average score presented in Sect. 4.2, we draw additional insights from comparing mode performance across all models and datasets available. Fig. 5 depicts the stacked bar plot of weighted F1-Scores averaged across all datasets and models for each augmentation that was available for the dataset. Across all the experiments, affine transformations, moving least squares (MLS) and random rotation DA techniques performed the

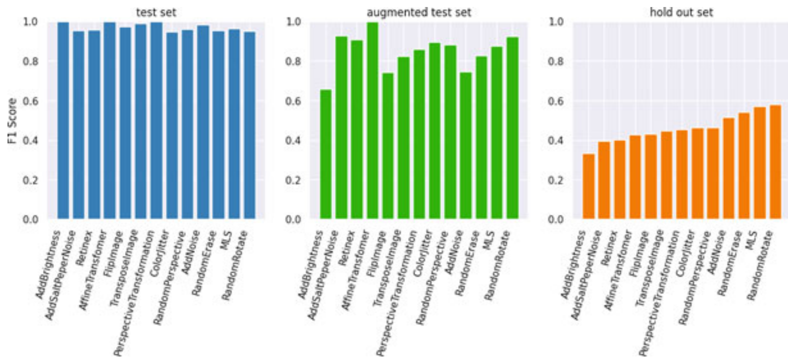


Fig. 5 F1-Scores averaged across all augmentations steps in the train sets

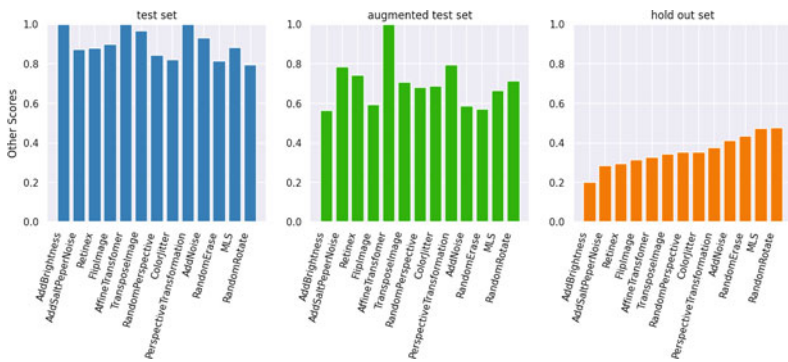


Fig. 6 Jaccard, precision, recall, kappa and MCC scores averaged across all augmentations steps in the train sets

best. Similarly, Fig. 6 depicts the average of the scores across all other evaluation metrics. We can observe similar trends where on average across experiments affine transformations, perspective transformation and MLS perform the best.

5 Conclusion

DL offers enormous potential to automate complex visual quality control tasks that cannot be solved using rule-based methods. However, manufacturing applications entail three severe data challenges: data sparsity, data imbalance and data shift. DA-methods have become an integral part of DL-pipelines to improve both performance and generalization. To provide precise assistance for the selection of DA-methods for developing DL-based quality control in the future, in this paper we present an experiment protocol. Thereby, we aim to evaluate the impact of individual DA-methods on defect detection performance

depending on dataset characteristics. We apply this protocol to three defect detection use-cases, present and interpret the results.

Using our approach, we can evaluate the influences of each DA method on the model metrics in detail. We show how to determine the domain shift between genuine and augmented dataset derivatives and therefore providing a measure and interpretability for choosing the degree of DA. By correlating this domain shift with F1-Scores, the strength of the positive influence of a DA-pipeline on bridging the domain shift can be determined. Applying our protocol to the datasets, we obtain the three best DA-methods MLS, Gaussian noise, random rotating (AITEX), image transpose, random perspective, salt & pepper noise (MagTile), and affine transformation, perspective transformation, image transpose (TIG5083). Thereby we confirm that the performance improvement of DA-methods depends on dataset characteristics, the DL-task to be solved and the degree of DA. This shows that there is no one-fits-all solution, but at the same time makes it all the more clear that establishing a mapping between dataset properties (e.g., degree of imbalance, defect sizes, positional variance of defects) and DA-induced performance enhancement will enable tailor-made and precise DL-pipeline development, especially in real-world applications.

Correlating the found performances with the respective domain shift revealed additional insights. The two pipelines for the AITEX dataset that induced the weakest negative correlation between domain shift and performance were mainly composed of our three best-performing augmentations for that dataset (see Table 5 pipeline 4,9). Additionally, we found that the worst performing pipelines either had very few augmentations or contained badly performing augmentations in them (mainly "random rotate" for AITEX), further highlighting the need for tailor-made DA-pipelines for each dataset. Our ablation study showed that (in contrast), by averaging the results over all datasets and models, at least some augmentations do perform better than others **on average**. The better-performing augmentations are the more complex ones, showcasing their versatility and robustness, while simple of-the-shelf augmentations display the least amount of lift in model performance. Fig. 6 can serve as a benchmark of augmentation techniques for new industrial-grade datasets, or those with unknown properties.

With the proposed the protocol, we lay the foundation for determining the appropriateness of DA-methods for specific data properties in an analytical approach. We will include also more advanced DA-methods and extend the study to additional domain-specific datasets to provide more validity to the results. By establishing a catalog of dataset properties to which we can map the results of the study, we aim to develop a domain-specific decision support system for choosing optimal DA-pipelines for DL-applications.

6 Appendix

6.1 Dataset Illustrations

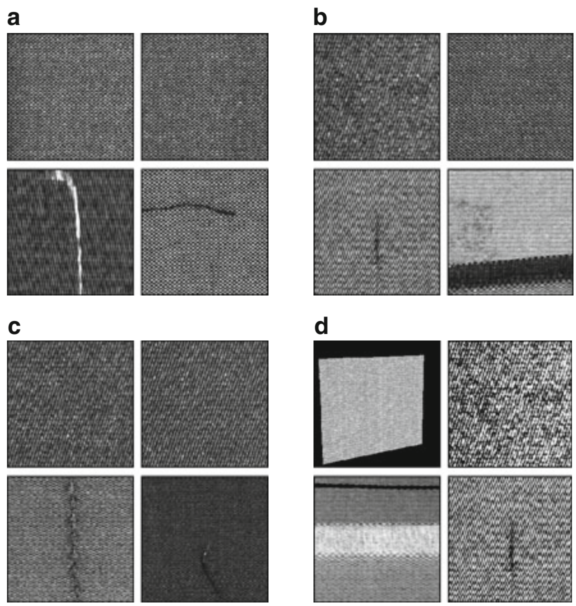


Fig. 7 Selection of AITEX [1] images: train set (**a**), test set (**b**), hold-out set (**c**), and augmented test set (**d**)

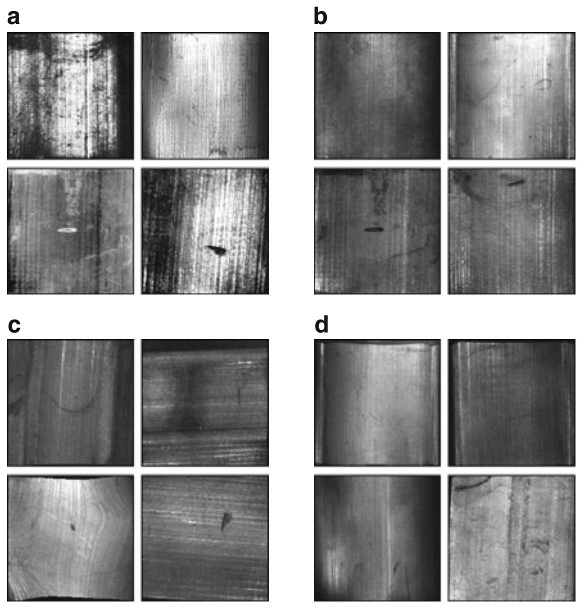


Fig. 8 Selection of MagTile [31] images: train set (**a**), test set (**b**), hold-out set (**c**), and augmented test set (**d**)

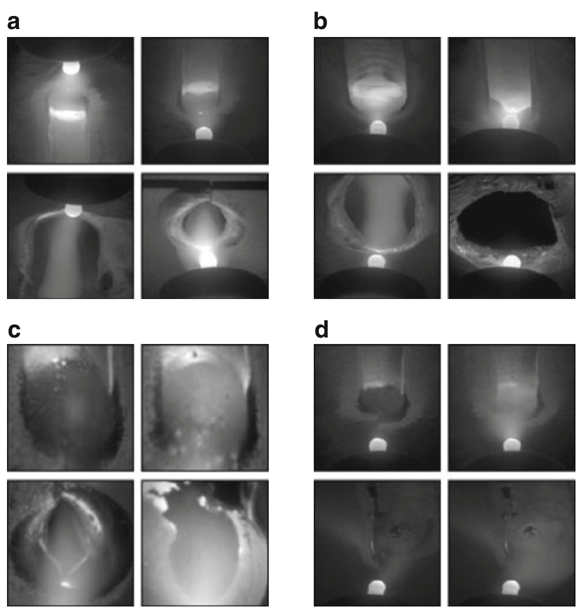


Fig. 9 Selection of TIG5083 [32] images: train set (**a**), test set (**b**), hold-out set (**c**), and augmented test set (**d**)

6.2 Domain Shift Calculations

The distance measure does not have good interpretability alone. Hence, we correlate the distance measure to the F1-Scores, a negative correlation is expected between them where the distance should be smaller, and the F1-Scores should be higher. Table 6 provides the distance measures for the averagepool layer of the ResNet-50 model across train and test sets, where the first three columns represent the distance and the following three columns represent the F1-score for the same pipelines. We take Pearson correlations along each pipeline, correlating the distance measure with the corresponding performance metric. Similarly, repeating this process for the last layers of both the models gives us Table 5. The same procedure was followed to construct similar tables for MagTile defects and TIG5083 dataset. Furthermore, we take the mean across the last layers of the models.

Table 6 Wasserstein distance between the augmented train set and all test sets for ResNet-50 and corresponding model F1-Scores

Pipeline	Averagepool layer ResNet-50(Distance)			Test results on ResNet-50		
	Train/Test	Train/Aug_test	Train/Hold_out	Test	Aug_test	Hold_out
1	0.0314	0.04008	0.13468	0.93218	0.91231	0.36298
2	0.30067	0.29725	0.72971	0.94262	0.81365	0.56211
3	0.05871	0.02578	0.14406	0.93388	0.92921	0.56211
4	0.01909	0.09311	0.0724	0.95699	0.90933	0.56211
5	0.07192	0.03547	0.16015	0.95217	0.92163	0.77694
6	0.09642	0.05205	0.16251	0.92431	0.92431	0.36298
7	0.04293	0.09987	0.16797	0.94673	0.87684	0.36298
8	0.02218	0.02751	0.03917	0.92921	0.92262	0.36298
9	0.03592	0.06176	0.0556	0.94262	0.92431	0.6417
10	0.0843	0.07488	0.17482	0.94046	0.89822	0.36298

6.3 Results

6.3.1 MagTile Dataset

Table 7 Train, validation and test set DA-Pipelines (MagTile)

Nr.	Train & validation Augmentation	Test Augmentation
1	Color Jitter, Salt & Pepper Noise	Flip Image, Transpose Image
2	Random Perspective, Flip Image	Salt & Pepper Noise, Retinex
3	Retinex, MLS	Salt & Pepper Noise, Random Perspective
4	Transpose Image, Random Perspective	MLS, Retinex
5	Color Jitter, Retinex, Salt & Pepper Noise	MLS, Flip Image
6	MLS	Retinex, Salt & Pepper Noise
7	Retinex, Color Jitter, MLS	Flip Image, Transpose Image
8	Random Perspective	MLS, Flip Image
9	Transpose Image	Random Perspective, Flip Image
10	Salt & Pepper Noise, Flip Image, Random Perspective, Retinex	MLS, Transpose Image

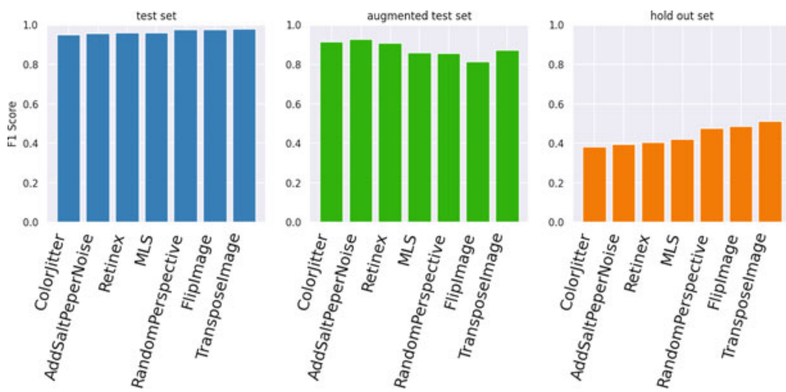


Fig. 10 F1-Scores averaged across models for each DA-method (MagTile). Sorted by hold-out performance.

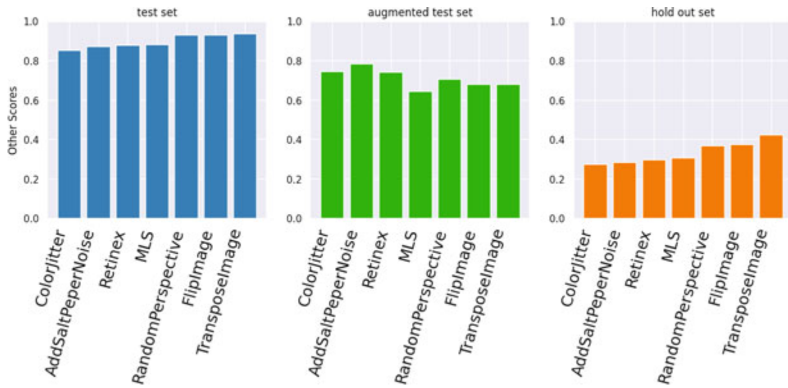


Fig. 11 Jaccard, precision, recall, kappa and MCC scores averaged across models for each DA-method (MagTile). Sorted by hold-out performance.

Table 8 Pearson correlations between the domain shift and model F1-Scores (MagTile). The bold values represent the largest negative mean correlations value.

Pipe-line	Inception v3			ResNet-50			Mean
	Last layer	2nd Last layer	3rd Last layer	Last layer	2nd Last layer	3rd Last layer	
1	−0.42134	−0.29361	−0.17805	−0.908	−0.92141	−0.99612	−0.61976 ± 0.3308
2	−0.87905	−0.77159	−0.75817	−0.98297	−0.91635	−0.98849	−0.88277 ± 0.09151
3	−0.99165	−0.86321	−0.89364	−0.99371	−0.95432	−0.97724	−0.94563 ± 0.05
4	−0.07302	−0.64191	−0.95817	0.86934	−0.99643	−0.63085	**−0.40517 ± 0.64512**
5	−0.99206	−0.99545	−0.99109	−0.99631	−0.99184	−0.99959	−0.99439 ± 0.00302
6	−0.27443	−0.28848	−0.41971	−0.85509	−0.91592	−0.90101	**−0.60911 ± 0.28593**
7	−0.99722	−0.99293	−0.9993	−0.40914	−0.5799	−0.98809	−0.82776 ± 0.24077
8	−0.98136	−0.87428	−0.90671	−0.71458	−0.83115	−0.99136	−0.88324 ± 0.09408
9	−0.98295	−0.93676	−0.82706	−0.94623	−0.96887	−0.9424	−0.93404 ± 0.05046
10	−0.98475	−0.97233	0.05444	−0.98154	−0.99635	−0.9711	−0.8086 ± 0.38606
	−0.7578 ± 0.3574	−0.7631 ± 0.2715	−0.6877 ± 0.3741	−0.6918 ± 0.5782	−0.9073 ± 0.1258	−0.9386 ± 0.1123	

6.3.2 TIG5083 Dataset

Table 9 Train, validation and test set DA-Pipelines (TIG5083)

Nr.	Train & validation Augmentation	Test Augmentation
1	Add Brightness	Affine Transfomer, Perspective Transformation
2	Add Brightness, Gaussian Noise	Transpose Image, Affine Transfomer
3	Transpose Image, Perspective Transformation, Affine Transfomer	Flip Image, Gaussian Noise
4	Gaussian Noise, Perspective Transformation	Transpose Image, Flip Image
5	Transpose Image, Affine Transfomer, Add Brightness	Gaussian Noise, Flip Image
6		Transpose Image, Add Brightness
7	Gaussian Noise, Transpose Image	Perspective Transformation, Flip Image
8	Gaussian Noise	Add Brightness, Affine Transfomer
9	Transpose Image, Add Brightness, Flip Image	Perspective Transformation, Gaussian Noise
10.	Perspective Transformation, Gaussian Noise	Flip Image, Transpose Image

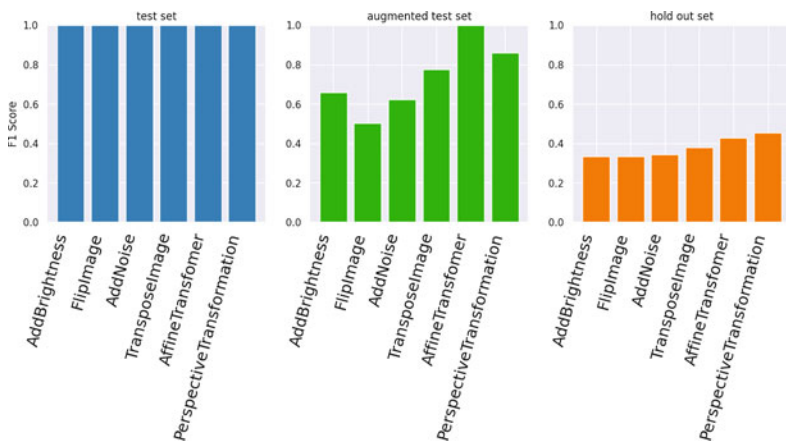


Fig. 12 F1-Scores averaged across models for each DA-method (TIG5083). Sorted by hold-out performance.

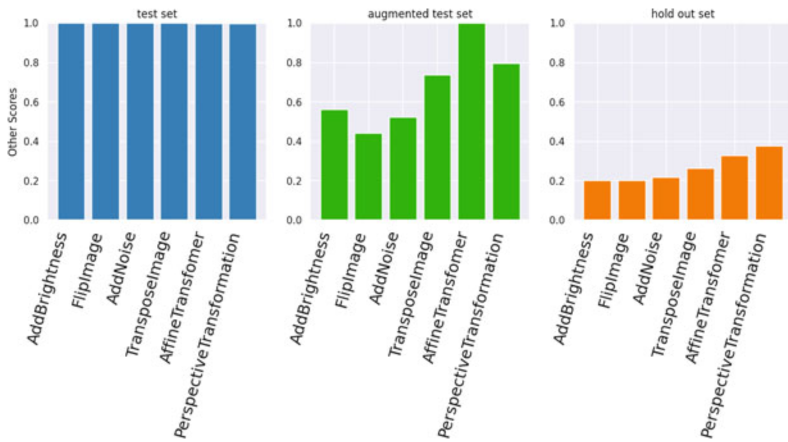


Fig. 13 Jaccard, precision, recall, kappa and MCC scores averaged across models for each DA-method (TIG5083). Sorted by hold-out performance.

Table 10 Pearson correlations between the domain shift and model F1-Scores (TIG5083). The bold values represent the largest negative mean correlations value.

Pipe-line	Inception v3			ResNet-50			Mean
	Last layer	2nd Last layer	3rd Last layer	Last layer	2nd Last layer	3rd Last layer	
1	−0.81792	−0.81945	−0.82353	−0.82422	−0.85945	−0.83561	−0.83003 ± 0.01433
2	−0.95836	−0.95362	−0.96833	−0.99884	−0.9493	−0.93576	−0.9607 ± 0.01966
3	−0.92238	−0.90818	−0.24479	−0.91736	−0.9626	−0.90916	−0.81074 ± 0.25376
4	−0.61701	−0.84784	−0.78521	−0.7663	−0.8051	−0.85062	**−0.77868 ± 0.07852**
5	−0.99923	−0.96855	−0.93786	−0.99872	−0.98831	−0.98185	−0.97909 ± 0.02119
6	−0.76129	−0.76871	−0.8904	−0.87209	−0.82126	−0.8541	−0.82798 ± 0.04921
7	−0.98311	−0.98905	−0.97248	−0.47366	−0.33301	−0.36318	**−0.68575 ± 0.29891**
8	−0.9528	−0.96702	−0.93089	−0.95038	−0.99583	−0.96655	−0.96058 ± 0.01985
9	−0.80877	−0.87399	−0.75693	−0.7757	−0.78879	−0.83407	−0.80638 ± 0.03882
10	−0.77332	−0.98553	−0.73991	−0.95172	−0.90796	−0.94447	−0.88382 ± 0.09322
	−0.8594 ± 0.1236	−0.9082 ± 0.0773	−0.805 ± 0.2152	−0.8529 ± 0.1579	−0.8412 ± 0.1942	−0.8475 ± 0.1789	

References

1. Silvestre-Blanes J, Albero-Albero T, Miralles I, Pérez-Llorens R, Moreno J (2019) A public fabric database for defect detection methods and results. Autex Res J 19(4):363–374. https://doi.org/10.2478/aut-2019-0035
2. Yang J, Li S, Wang Z, Dong H, Wang J, Tang S (2020) Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges. Materials 13(24):5755. https://doi.org/10.3390/ma13245755
3. Minhas MS, Zelek JS (2020) Defect detection using deep learning from minimal annotations. In: Farinella GM, Radeva P, Braz J (Hrsg) Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2020, Volume 4: VISAPP, Valletta, Malta. SCITEPRESS, Setúbal, S 506–513 https://doi.org/10.5220/0009168005060513
4. Hssayeni M, Saxena S, Ptucha R, Savakis A (2017) Distracted driver detection: Deep learning vs handcrafted features. Electron Imaging 2017:20–26. https://doi.org/10.2352/ISSN.2470-1173.2017.10.IMAWM-162
5. Marnissi MA, Fradi H, Dugelay JL (2019) On the discriminative power of learned vs. handcrafted features for crowd density analysis. In: 2019 International Joint Conference on Neural Networks (IJCNN), S 1–8 https://doi.org/10.1109/IJCNN.2019.8851764
6. Sun C, Shrivastava A, Singh S, Gupta A (2017) Revisiting unreasonable effectiveness of data in deep learning era. http://arxiv.org/pdf/1707.02968v2
7. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 770–778 https://doi.org/10.1109/CVPR.2016.90
8. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 2818–2826 https://doi.org/10.1109/CVPR.2016.308
9. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, S 248–255 https://doi.org/10.1109/CVPR.2009.5206848
10. Lin T, Maire M, Belongie SJ, Bourdev LD, Girshick RB, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. CoRR. http://arxiv.org/abs/1405.0312
11. LeCun Y, Cortes C (2010) MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/
12. Zhong Z, Zheng L, Kang G, Li S, Yang Y (2020) Random erasing data augmentation. AAAI 34(07):13001–13008
13. Zhou K, Liu Z, Qiao Y, Xiang T, Loy CC (2021) Domain generalization in vision: a survey (arXiv e-prints arXiv:2103.02503)
14. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
15. Weiss K, Khoshgoftaar T, Wang D (2016) A survey of transfer learning. J Big Data. https://doi.org/10.1186/s40537-016-0043-6
16. Erhan D, Bengio Y, Courville A, Manzagol PA, Vincent P, Bengio S (2010) Why does unsupervised pre-training help deep learning? J Mach Learn Res 11(19):625–660
17. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6:60. https://doi.org/10.1186/s40537-019-0197-0
18. Feng SY, Gangal V, Wei J, Chandar S, Vosoughi S, Mitamura T, Hovy E (2021) A survey of data augmentation approaches for NLP. In: Findings of the Association for Computational Lin-

guistics: ACL-IJCNLP 2021. pp. 968–988. Association for Computational Linguistics. https://aclanthology.org/2021.findings-acl.84
19. Iwana BK, Uchida S (2021) An empirical survey of data augmentation for time series classification with neural networks. PLoS ONE 16(7):e254841
20. Wan C, Shen X, Zhang Y, Yin Z, Tian X, Gao F, Huang J, Hua XS (2022) Meta convolutional neural networks for single domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), S 4682–4691
21. Qiao F, Zhao L, Peng X (2020) Learning to learn single domain generalization. CoRR. https://arxiv.org/abs/2003.13216
22. Xu Z, Liu D, Yang J, Niethammer M (2020) Robust and generalizable visual representation learning via random convolutions. CoRR. https://arxiv.org/abs/2007.13003
23. Faryna K, van der Laak J, Litjens G (2021) Tailoring automated data augmentation to h&e-stained histopathology. In: Medical imaging with deep learning. https://openreview.net/forum?id=JrBfXaoxbA2
24. Zhang L, Wang X, Yang D, Sanford T, Harmon SA, Turkbey B, Roth H, Myronenko A, Xu D, Xu Z (2019) When unseen domain generalization is unnecessary? rethinking data augmentation. CoRR. http://arxiv.org/abs/1906.03347
25. Jackson PTG, Abarghouei AA, Bonner S, Breckon TP, Obara B (2018) Style augmentation: Data augmentation via style randomization. CoRR. http://arxiv.org/abs/1809.05375
26. Meister S, Wermes MAM, Stüve J, Groves RM (2021) Review of image segmentation techniques for layup defect detection in the automated fiber placement process. J Intell Manuf 32(8):2099–2119
27. Shijie J, Ping W, Peiyi J, Siping H (2017) Research on data augmentation for image classification based on convolution neural networks. In: 2017 Chinese Automation Congress (CAC), S 4165–4170 https://doi.org/10.1109/CAC.2017.8243510
28. Yang S, Xiao W, Zhang M, Guo S, Zhao J, Shen F (2022) Image data augmentation for deep learning: A survey. http://arxiv.org/pdf/2204.08610v1
29. Khosla C, Saini BS (2020) Enhancing performance of deep learning models with different data augmentation techniques: A survey. In: 2020 International Conference on Intelligent Engineering and Management (ICIEM). IEEE, New York, S 79–85 https://doi.org/10.1109/ICIEM48762.2020.9160048
30. Jain S, Seth G, Paruthi A, Soni U, Kumar G (2022) Synthetic data augmentation for surface defect detection and classification using deep learning. J Intell Manuf 33(4):1007–1020. https://doi.org/10.1007/s10845-020-01710-x
31. Huang Y, Qiu C, Guo Y, Wang X, Yuan K (2018) Surface defect saliency of magnetic tile. In: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), S 612–617 https://doi.org/10.1109/COASE.2018.8560423
32. Bacioiu D, Melton G, Papaelias M, Shaw R (2019) Automated defect classification of aluminium 5083 tig welding using hdr camera and neural networks. J Manuf Process 45:603–613
33. Schaefer S, McPhail T, Warren J (2006) Image deformation using moving least squares. ACM Trans Graph 25(3):533–540. https://doi.org/10.1145/1141911.1141920
34. Petro AB, Sbert C, Morel JM (2014) Multiscale retinex. Image Process Line 4:71–88. https://doi.org/10.5201/ipol.2014.107
35. Jobson D, Rahman Z, Woodell G (1997) A multiscale retinex for bridging the gap between color images and the human observation of scenes. IEEE Trans Image Process 6(7):965–976. https://doi.org/10.1109/83.597272
36. Stacke K, Eilertsen G, Unger J, Lundström C (2021) Measuring domain shift for deep learning in histopathology. IEEE J Biomed Health Inform 25(2):325–336. https://doi.org/10.1109/JBHI.2020.3032060

37. Hancock J (2004) Jaccard distance (Jaccard index, Jaccard similarity coefficient) https://doi.org/10.1002/9780471650126.dob0956
38. McHugh M (2012) Interrater reliability: The kappa statistic. Biochem Med 22:276–282. https://doi.org/10.11613/BM.2012.031
39. Chicco D, Jurman G (2020) The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC Genomics. https://doi.org/10.1186/s12864-019-6413-7
40. Loshchilov I, Hutter F (2019) Decoupled weight decay regularization. In: ICLR

Creating Synthetic Training Data for Machine Vision Quality Gates

Iris Gräßler, Michael Hieb, Daniel Roesmann, and Marc Unverzagt

Abstract

Manufacturing companies face the challenge of combining increasing productivity and quality standards with customer-oriented mass production. To achieve the required quality standards, quality controls are carried out after selected production steps. These are often visual inspections by trained personnel based on checklists. To automate visual inspection industrial, cameras and powerful machine vision algorithms are needed. Large amounts of visual training data are usually required in order to train these algorithms. However, collecting training data is time-consuming, especially in customer-oriented mass production. Synthetic training data generated by CAD tools and rendering software can alleviate the lack of available training data. Within the paper at hand, a novel approach is presented examining the use of synthetic training data in machine vision applications. The results show that synthetically generated training data used to train machine vision quality gates is fundamentally suitable. This offers great potential to relieve process and productions developers in the development of quality gates in the future.

I. Gräßler · M. Hieb (✉) · D. Roesmann · M. Unverzagt
Heinz Nixdorf Institute, Paderborn, Deutschland
e-mail: iris.graessler@hni.uni-paderborn.de

M. Hieb
e-mail: michael.hieb@hni.uni-paderborn.de

D. Roesmann
e-mail: daniel.roesmann@hni.uni-paderborn.de

M. Unverzagt
e-mail: marc.unverzagt@hni.uni-paderborn.de

V. Lohweg (Hrsg.), *Bildverarbeitung in der Automation*, Technologien für die intelligente Automation 17, https://doi.org/10.1007/978-3-662-66769-9_7

Keywords

Production control · Synthetic training data · Machine vision quality gate · Synthetic data in mass customization · Production planning

1 Introduction

Within mass customization, manufacturing companies face the challenge of combining productivity increases and high-quality standards. Quality controls are carried out after selected production steps to achieve the required quality standards. They represent measuring points within production for checking quality-relevant product properties. Today, quality control is often a visual inspection by trained personnel using checklists. For example, in the assembly of small electronic products, a worker checks that the necessary screws haven been fitted and that the component has been properly assembled. These manual checks are time-consuming and cost intensive. Also, they are characterized by a high degree of monotony in the form of repetitive processes with a fixed sequence and are prone to errors [1].

Especially visual inspections offer a high potential for automation. Advances in deep learning show innovative possibilities for industrial production. Automated visual inspections in production can be carried out using machine vision. The quality gates consist of one or several cameras, light source, trigger, production line control and image processing software [2]. However, training the image processing software using deep learning methods requires large amounts of data in order to perform.

The basis for a successful implementation of deep learning methods is high-quality annotated training data [2]. Current approaches use extensive amounts of annotated real image datasets for this purpose. Especially, data collection of image datasets are one of the most time-consuming and cost-intensive steps in the implementation of deep learning methods [3]. Synthetically generated training image data provide a remedy. Training with synthetic data offers many advantages for automated quality control especially in mass customization with high number of variants since unlimited amounts of data can be produced [4]. Within object detection approaches to apply synthetic data have been developed in recent years see [5–7] . However, the application of synthetic data for machine vision quality gates in automation is currently a research gap.

The aim of this paper is to investigate the use of synthetic training data for machine vision quality gates in mass customization. Using CAD and rendering software, three different training datasets are generated: 1) the first training dataset forms the baseline and consists entirely of real image data; 2) the second training dataset contains exclusively synthetic training data; 3) the training dataset is hybrid containing 95% synthetic and 5% real image data. For the comparison of the three approaches, Accuracy, Precision, Recall and F1-Score are used as evaluation criteria.

For validation purposes, the assembly of the open-source jointed-arm robot “Zortrax”, which is produced in the Smart Automation Laboratory of the Heinz Nixdorf Institute, serves as an example [7]. Within this work one assembly step of the Zortrax robot is evaluated using deep learning methods trained on synthetic image data. The research contribution is intended to evaluate the fundamental question of whether synthetically generated image data is generally suitable for machine vision quality gates.

2 State of the Art

Recently, synthetic data is applied in the context of improving the performance of object detection algorithms (see [5, 7]). Using synthetic data in the context of industrial production and for training machine vision quality gates still represents a research gap.

2.1 Machine Vision Quality Gates

Machine vision quality gates document the product quality and trace products throughout the production. Usually, the quality of products is evaluated after defined production steps, which are critical for the product quality. Therefore, assembly stations, machines or productions lines are extended using cameras to perform visual quality inspections in order to remove or rework defective products [2].

In order to fulfill visual quality inspections different abilities are needed. Some common tasks of machine vision quality gates are [8, 9]: (1) counting pixels, (2) template matching, (3) segmentation, (4) barcode reading, (5) object identification, (6) position detection, (7) completeness checks, (8) shape and dimensional inspections, (9) surface inspection. The typical hardware components of machine vision quality gates are (see Fig. 1; [2]):

(1) **Programmable Logic Controller** (PLC) which steers the production process
(2) **Trigger** is used to active the image acquisition process over the camera(s)
(3) **Conveyer Belt** mechanically transports the products through the productions
(4) **Illumination** of the object using special designed lightning ensuring a high image quality
(5) **(Smart) Camera(s)** the object is imaged using lenses and light sensors
(6) **(Edge) Computer** directly build into the camera or standalone classifies the acquired image
(7) **Machine Vision Software** inspects the image and returns an evaluation to the PLC (for example MVTec Halcon)

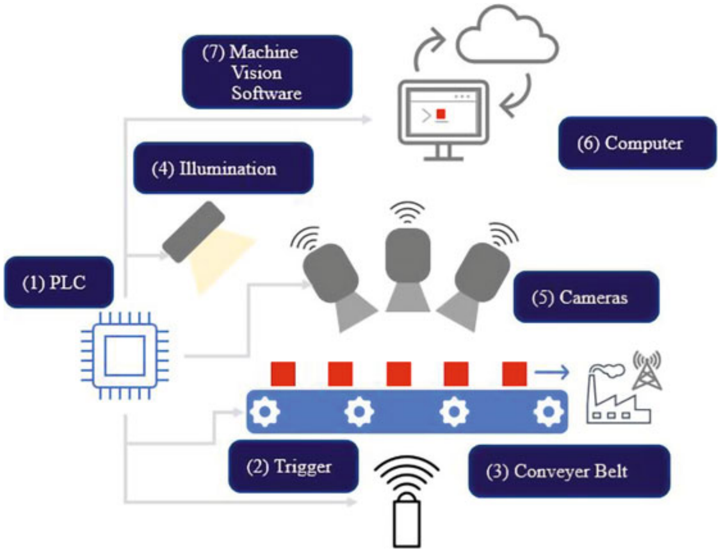


Fig. 1 Components of machine vision quality gates (based on: [9]).

However, depending upon the task machine vision quality gates might differ slightly. Visual inspection on manual assembly stations usually do not require a mechanically transportation and the triggers are controlled over buttons/foot pedals.

2.2 Synthetic Training Data

Synthetic training data has already been used in computer vision approaches. Applications are object detection [5, 7] , classification [6] or segmentation [10, 11]. Within the application areas, it is either impossible to collect labelled training data manually, or impossible to obtain data. In these cases, synthetic trainings show great success.

Existing approaches that are trained using real image data go through the following seven steps, which optionally iterate if key performance indicators are not reached: (1) data acquisition, (2) data cleaning, (3) data annotation, (4) model training, (5) model testing and (6) model deployment (see Fig. 2). Within the first step data are acquired from imaging devices like a camera (typically captured as images or sequences from videos). Secondly, each image data is pre-processed for the purpose of standardization, these include resizing, blurring, rotating, etc. Thirdly, the pre-processed image data is annotated assigning metadata in form of classes or key to the image. Furthermore, the selected DL model is trained. Within the next step the model is analyzed using test data. Usually, test data is excluded from model training ensuring the validity of the trained model. Additionally, key performance indicators are used to evaluate the performance.

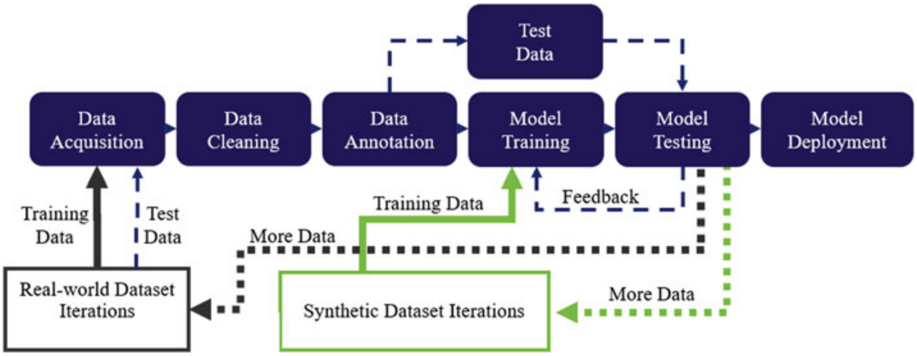


Fig. 2 Comparison of real-world dataset iterations vs. synthetic dataset iterations

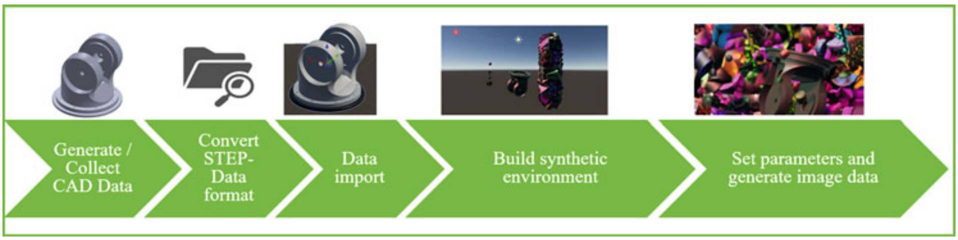


Fig. 3 Synthetic training data generation pipeline (based on [5]).

If the defined key performance indicators conclude poor results possibly training parameters needed to be improved, or more training data is needed. Lastly, if the key performance indicators reach the determined quality level the model can be exported and used for deployment.

The steps of data acquisition, data cleaning and data annotation are the most time-consuming and cost-intensive steps in the implementation of DL algorithms [3]. Synthetic data can bypass these steps (see Fig. 3) and reduce the implementation time significantly. Additionally, synthetic training data does not need the physical object, which enable fast data generation for multiple tasks. Bridging the domain gap using synthetic training data can be executed by simply cutting out relevant objects from real images and mapping them onto random selected background as shown in [11]. This approach bridges the domain gap by using images from real domains, which therefore are able to bridge the domain gap. The drawback of this approach is clearly the missing possibility to generate synthetic training images from different perspectives and light conditions, which states a limitation.

Domain Randomization (DR) was introduced in [4], enabling synthetic images from different perspectives and light conditions. This approach randomly creates a 3D environment using varieties of textures, numbers of light sources, color, background as well as foreground objects. The aim of DR is to bridge the domain gap using a sufficient number

of variations. The drawback of DR is that large amounts of training data is needed and that networks are unable to correctly identify small differences within similar classes.

The Unity Perception package introduced by [5] builds upon DR enabling customizable image data generation out of the box including ground truth annotations. The package supports 2D/3D object detection, semantic segmentation, instance segmentation, and keypoints estimation. Several settings make it possible generating millions of annotated image data.

3 Synthetic Training Data Generation for Machine Vision Quality Gates

Within this work, the use of synthetic training data in the context of machine vision quality gates is explored. Therefore, a pipeline (see Fig. 4) is introduced using the Unity Perception package to generate synthetic training data in the context of machine vision quality gates in mass customization.

3.1 Synthetic Data Generation Pipeline for Classification

Within this section, the pipeline of creating synthetic annotated training data for classification is explained in detail. The basis for the pipeline is Domain Randomization. Creating synthetic annotated training datasets for classification tasks requires five basis steps (see Fig. 4): (1) generate/collect CAD data, (2) convert data format, (3) data import, (4) build environment (5) set parameters and generate image data.

The first step to create synthetic image data is to generate (e.g., using Solidworks) or collect CAD data. In the context of industrial production, CAD data usually is created within product creation. The required level of details of CAD data depends upon the subsequent application for which the image data will be used. Missing individual parts inevitably lead to fails bridging the domain gap.

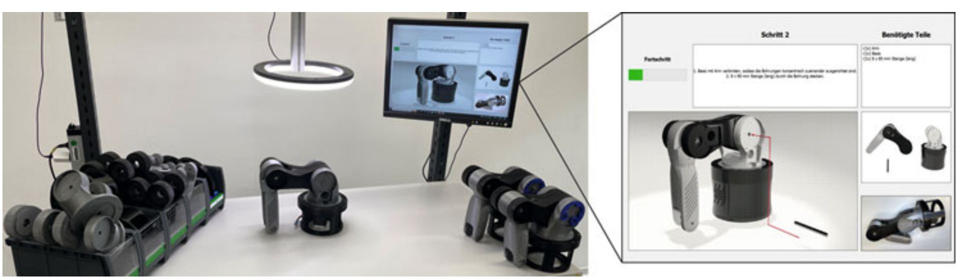


Fig. 4 Assembly stations equipped with graphical user interface and machine vision quality gate.

After the CAD data is acquired, the data format needs to be converted into Unity friendly formats, for example Filmbox (.fbx). A software tool able to convert STEP to Filmbox is Autodesk 3DS MAX or Blender. Subsequently, the converted Filmbox data is imported into the Unity software environment.

In the third step, the 3D environment is created. Therefore, the Unity Perception packet is used [5]. To create a 3D environment capable of bridging the domain gap several aspects needed to be considered. According to [5], the 3D environment need to contain the classified object, background noise objects, occluding noise objects, and randomized object textures. These settings are programmed within the Unity Perception package. The result is presented in Fig. 3. There are three layers of objects creating the necessary variation into the synthetic environment.

Lastly, the environmental parameters of randomized object color, lighting parameters, camera post and camera movement are programmed. As soon as the 3D environment is set, 2D images can be captured out of the synthetic 3D scene. Additionally, each captured image is labeled automatically.

4 Validation

Within this section, detailed experiment settings and results are reported. Furthermore, validation settings are documented which were used to train deep neural networks in the context of production. Finally, the chapter summarizes validation results.

4.1 Smart Automation Laboratory

The Smart Automation Laboratory located at the Heinz Nixdorf Institute (Paderborn, Germany) is used for research in the fields of production planning and control as well as automation technology, such as machine vision quality gates. The laboratory consists of three manufacturing cells, a material flow system and an adaptive assembly workstation (see Fig. 4). The manufacturing cells and the assembly workstation are connected via the rail-bound material flow system. Shuttles are used on the rails, which receive orders from the ERP system and process them in communication with the manufacturing cells and the assembly station [12, 13] . Currently, the open-source robot arm Zortrax is manufactured at the laboratory, which consist of 34 separate part and require 29 assembly steps [13].

The assembly workstation is equipped with boxes for parts, a camera including lighting source and an assistant system. The assistant system visualize status, assembly description, required parts, animated video and real time camera cast (see Fig. 4). The real time camera cast is used to evaluate the process and the quality of the assembly process. Furthermore, the classification is done using deep neural networks, which were initially trained using real image data. Within this research these deep neural networks are trained using synthetic training images and validated using real image data.

4.2 Validation Setting

Within the validation, the aim was to properly classify one sequential assembly step of the robot arm Zortrax (see Fig. 5). The aim of the quality inspection was to properly evaluate the assembly step 12 using synthetically trained DL models.

As visible in Fig. 5 the assembly step needs various parts and includes several work steps. First, the ring needs to be placed under the carrier. Subsequently, the tooth ring needs to be placed under the carrier and the ring to be correctly assembled. Forgetting the ring in between the carrier and tooth ring leads to a quality deficiency (wrong assembly, see Fig. 5). If the parts are placed in proper order the carrier is attached to the tooth ring using eight screws and nuts (correct assembly, see Fig. 5). This forms a binary classification problem to correctly supervise the assembly step 12.

Three datasets were acquired and annotated (see Table 1). All trained DL models were evaluated using the same real test and validation image data. The first out of three-training dataset contains 1000 real images per class (i.e., assembly step 12) captured using a standard camera and ring illumination. The second training dataset purely contains synthetically generated image data using the previous explained pipeline. The last dataset is a hybrid dataset containing 50 real images and 950 synthetically generated images per class. This was done based on the results of [3] finding out that already small percentages

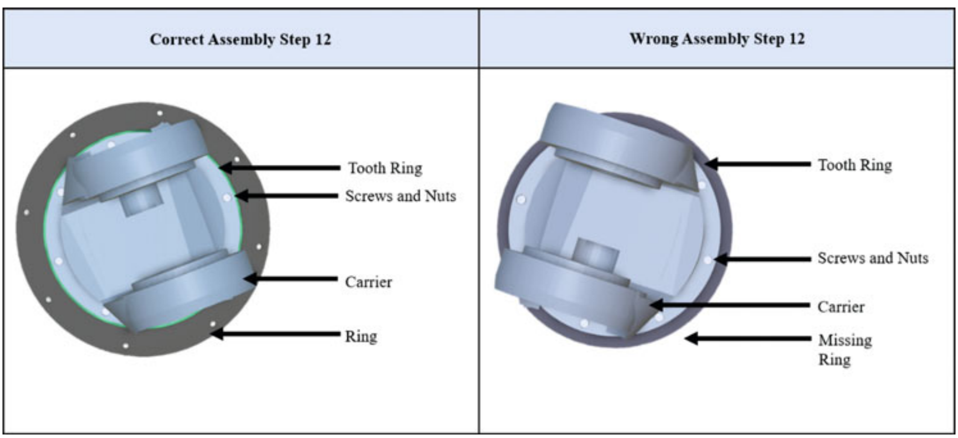


Fig. 5 Classification problem of assembly step 12.

Table 1 Summary of training, test and validation datasets.

Datasets	Training () = amount of synthetic data	Test	Validation
Real dataset	2000	100	100
Synthetic dataset	2000 (2000)	100	100
Hybrid dataset	2000 (100)	100	100

Table 2 Used key performance indicators (based on [2])

Metric	Description	Formulas
Precision	Calculated by dividing the number of true predictions by the number of true positives plus false positives	$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$
Recall	Calculated dividing the number of true predictions by the number of true positives plus false negatives	$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$
F1 Score	Is the harmonic mean between precision and recall	$\text{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recal}}{\text{precision} + \text{recall}}$

of real image data in synthetically generated image data boost the performance of DL algorithms.

Within this work three DL algorithms were trained for classification. These are: (1) DenseNet 201 [14], (2) ResNet 152 v2 [15] and (3) Xception [16]. Each model was trained using the above listed training-, test- and validation datasets with the target size 224×224. All models were trained with the optimizer Adam, 0.001 learning rate and 10 epochs. Table 1 summaries the distribution of image data used for training.

4.3 Key Performance Indicators

Within the validation the models were evaluated using the following key performance indicators (see Table 2). This was done to evaluate trained DL models. The key performance indicators are used as measurements for comparison and validation. The most common metrics for supervised learning are summarized in Table 2.

4.4 Results & Discussion

In this section the performance of trained DL models using synthetic training data is compared to real-world training data. Table 3 displays the results for each of the trained DL models. The results show that the performance of synthetic training data generally can bridge the domain gap. The quality of assembly step 12 can moderate been inspected using synthetic generate training data.

The findings support the thesis that synthetically generated training data can be used for machine vision quality gates in production. Synthetically generated training data created using Domain Randomization and the Unit Perceptions package provide the ability to quickly generate training datasets. The outcome offers many advantages for automated quality control, especially in mass customization with high number of variants, since unlimited amounts of training data can be produced.

Table 3 Summary of results after 10 training epochs.

Datasets	Precision macro avg.	Recall macro avg.	F1 Score macro avg.
Synthetic dataset			
– DenseNet 201	0.87	0.86	0.86
– ResNet 152 v2	0.86	0.85	0.85
– Xception	0.89	0.89	0.89
Real dataset			
– DenseNet 201	1.00	1.00	1.00
– ResNet 152 v2	1.00	1.00	1.00
– Xception	1.00	1.00	1.00
Hybrid dataset			
– DenseNet 201	1.00	1.00	1.00
– ResNet 152 v2	0.99	0.99	0.99
– Xception	0.99	0.99	0.99

However, compared to the baseline it can be seen that a slight domain gap still exists between training with real and synthetic data. To further bridge the domain gap, improvements needed to be done. These include the synthetic data generation pipeline as well as the training settings. Additionally, supported by [5], more training data might be needed for training using synthetic datasets. Within their approach [5] several thousand image data were generate for one class.

Lastly, the evidence shows that using small amounts of real data (third training dataset) already improves the performance of the DL model significantly. Therefore, the results also support the thesis claimed by [5, 7] stating that small sets (5%–10%) of real data boost the performance of trained DL models.

5 Summary & Future Studies

The results show that synthetically generated training datasets are fundamentally suitable for training machine vision quality gates. This offers great potential to relieve process and production developers in the development of quality gates in the future. Synthetically generated image data seems to enable automatic quality controls for high number of product variants especially in the field of mass customization.

To conclude, domain randomization using the Unity perception package to create synthetic image data is a promising research direction to be future examined. This paper presents a solution for a classification model to inspect one assembly step of the robotic arm Zortrax trained on synthetic, hybrid and real image data at high accuracy. The trained models enable early error detection in the assembly process and are therefore able to ensure a high quality in the assembly. Since the chosen example is still a very simple binary classification problem, the results show very good key performance indicators, as expected. In order to further bridge the domain gap improvements might be needed.

Within in future studies, the pipeline to create synthetic image data needs to be further improved. Settings such as background and foreground objects, texture and different parameter setting need to be further examined. In this, an increase in performance and generalization is to be expected. Future work will explore how to make this technique reliable and effective to use in industrial production.

References

1. Gräßler I, Pöhler A (2019) Human-centric design of cyber-physical production systems. Procedia CIRP 84:251–256
2. Steger C, Ulrich M, Wiedemann C (2018) Machine vision algorithms and applications. Wiley-VCH, Weinheim
3. Hinterstoisser S, Olivier P, Hauke Marek HMBM (2019) An annotation saved is an annotation earned: using fully synthetic training for object detection
4. Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbel P (2017) Domain randomization for transferring deep neural networks from simulation to the real world. https://arxiv.org/pdf/1703.06907 (Created 03.2017)
5. Borkman S, Crespi A, Dhakad S, Ganguly S, Hogins J, Jhang Y-C, Kamalzadeh M, Li B, Leal S, Parisi P, Romero C, Smith W, Thaman A, Warren S, Yadav N (2021) Unity perception: generate synthetic data for computer vision. https://arxiv.org/abs/2107.04259 (Created 07.2021)
6. Borrego J, Dehban A, Figueiredo R, Moreno P, Bernardino A, Santos-Victor J (2018) Applying domain randomization to synthetic data for object category detection. https://arxiv.org/pdf/1807.09834 (Created 07.2018)
7. Gräßler I, Pöhler A (2018) Intelligent devices in a decentralized production system concept. Procedia CIRP 67:116–121
8. Labudzki R, Legutko S, Raos P (2014) The essence and applications of machine vision. Tehnicki Vjesnik 21(4):903–909
9. Atr, 3D-Vision: MVTec Software. https://www.mvtec.com/de/technologien/3d-vision
10. Ward D, Moghadam P, Hudson N (2018) Deep leaf segmentation using synthetic data. https://arxiv.org/pdf/1807.10931 (Created 07.2018)
11. Dwibedi D, Misra I, Hebert Cut M (2017) Paste and learn: surprisingly easy synthesis for instance detection. http://arxiv.org/pdf/1708.01642v1 (Created 08.2017)
12. Gräßler I, Pöhler A (2020) Produktentstehung im Zeitalter von Industrie 4.0. In: Handbuch Gestaltung digitaler und vernetzter Arbeitswelten. Springer, Berlin, Heidelberg, pp 383–403
13. Zortrax Library Zortrax robotic arm by Zortrax – Zortrax library. https://library.zortrax.com/project/zortrax-robotic-arm/
14. Wang S-H, Zhang Y-D (2020) Densenet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification. ACM Trans Multimed Comput Commun Appl 16(2s):1–19
15. TensorFlow https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet/ResNet152
16. K. Team Keras documentation: Xception. https://keras.io/api/applications/xception/